В. А. Гвоздева

ИНФОРМАТИКА, АВТОМАТИЗИРОВАННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ

Рекомендовано Научно-методическим советом Московского государственного института электронной техники (технического университета) в качестве учебника для студентов технических специальностей

MUHAMMAD AL-XORAZMIY NOMIDAGI
TOSHKENT AXBOROT
TEXNOLOGIYALARI UNIVERSITETI
387144
AXBOROT-RESURS MARKAZI

Электронно-
Библиотечная
Система
znanium.com

МОСКВА
ИД «ФОРУМ» — ИНФРА-М
2018

УДК 004(075.32)
ББК 32.81я723
Г25

ФЗ
№ 436-ФЗ

Издание не подлежит маркировке
в соответствии с п. 1 ч. 4 ст. 11

Рецензенты:

кандидат технических наук, доцент, заведующий кафедрой информатики и компьютерных технологий МГАВТ *М. И. Иванов*;
доктор технических наук, профессор кафедры информатики и программного обеспечения вычислительных систем Московского государственного института электронной техники *Л. Г. Гагарина*

Гвоздева В. А.

Г25 Информатика, автоматизированные информационные технологии и системы : учебник / В. А. Гвоздева. — М.: ИД «ФОРУМ»: ИНФРА-М, 2018. — 544 с.: ил. — (Профессиональное образование).

ISBN 978-5-8199-0449-7 (ИД «ФОРУМ»)
ISBN 978-5-16-004572-6 (ИНФРА-М, print)
ISBN 978-5-16-102403-4 (ИНФРА-М, online)

Учебник написан в соответствии с Федеральным государственным образовательным стандартом 3-го поколения. В первой части книги, «Информатика», даны история развития вычислительной техники и становления информатики, вопросы представления, измерения и хранения информации, системы счисления, логические основы, архитектура и устройства ЭВМ, основные понятия операционных систем и их файловая структура, системное и прикладное программное обеспечение. Особое внимание уделено основам знаний по алгоритмизации, технологии программирования, языкам программирования, а также системе объектно-ориентированного программирования MS Visual Basic. Во второй части книги, «Информационные технологии», излагаются вопросы компьютерной обработки текстовой, числовой, графической информации, основы баз данных и знаний, систем управления базами данных (СУБД), даются представление о локальных и глобальных компьютерных сетях и знания о средствах создания веб-документов. Третья часть книги, «Автоматизированные информационные системы», посвящена вопросам разработки и функционирования АИС. Рассматриваются вопросы необходимости автоматизации информационных потоков, состав и структура АИС, методы и стадии их разработки, обеспечивающая и функциональные части, типы АИС, тенденции развития информационных систем.

Для подготовки студентов вузов и колледжей, обучающихся по техническим специальностям.

УДК 004(075.32)
ББК 32.81я723

ISBN 978-5-8199-0449-7 (ИД «ФОРУМ»)
ISBN 978-5-16-004572-6 (ИНФРА-М, print)
ISBN 978-5-16-102403-4 (ИНФРА-М, online)

© Гвоздева В. А., 2015
© ИД «ФОРУМ», 2015

Введение

Современная цивилизация характеризуется тем, что наиболее развитая ее часть находится в переходе от индустриального к информационному обществу, построенному на использовании большого объема информации и знаний всеми его членами. Информация во многих организациях становится ключевым ресурсом, а информационная обработка — делом огромной важности. Современный уровень информатизации общества предопределяет использование новейших технических, технологических, программных средств, т. е. автоматизированных информационных технологий и систем.

Основным направлением современной информатики как науки является исследование методологии информационной деятельности в условиях широкого применения компьютерной и телекоммуникационной техники для конкретных приложений.

Исследования в теории информации, развитие интегральных информационных систем, информационных технологий, электронных библиотек, их объединение с системами управления документами и бизнес-процессами позволило создать новый тип сложных корпоративных, отраслевых и межотраслевых систем управления на основе активного сетевого взаимодействия в интернет/интранет-технологиях.

Изучение проблем в контексте исследований информации, информационных потоков, информационных технологий (ИТ), автоматизированных информационных систем (АИС), автоматизированных информационных технологий (АИТ), автоматизированных систем управления (АСУ) и т. п. и всего, что с ними связано, проводится Всероссийским институтом научной и технической информации Российской Академии наук (ВИНИТИ РАН). Он осуществляет тестирование автоматизированных информационных систем, формирует отчеты и отправляет их компании-производителю.

АИС представляют собой системы, основанные на постоянно развивающихся концепциях использования информации. Напри-

мер, сегодня информационно-поисковые системы (ИПС) являются наиболее мощным механизмом поиска сетевых информационных ресурсов. Наиболее крупные международные поисковые системы — Google, Yahoo!, MSN, OpenText, AltaVista и др., в русском Интернете (Рунете) — Яндекс, Rambler, Апорт. О крупных международных поисковых системах в российском Интернете см. разделы 1.1, 7.1.1, 10.1 и др.

В настоящее время поисковые системы представляют собой не только инструменты поиска информации, но и сферы для бизнеса.

Например, Yahoo!, созданная в 1994 г., — на сегодняшний день самый старый и наиболее полный каталог интернет-ресурсов — является одним из самых популярных поисковых средств, имеет базу данных более 1 млн проиндексированных сайтов. Yahoo! обладает большим количеством всевозможных сервисов:

- Yahoo.mail (почта) — бесплатный почтовый ящик для каждого желающего;
- Yahoo.messenger (сообщения)— клиент для общения в сети Интернет;
- Yahoo.radio (радио) — прослушивание радиопрограмм через Интернет;
- Yahoo.Answers (ответы) — любой зарегистрированный желающий может оставить свой вопрос на сайте, и другие пользователи смогут на него ответить;
- Yahoo.autos (автомобили) — огромный каталог автомобилей, как новинок, так и б/у. Здесь можно найти последние новости мирового автопрома, купить или продать автомобиль;
- Yahoo.Finance (финансы) — любая информация в сфере финансов. Новости, прогнозы, личные финансы и даже возможность разослать свое резюме;
- Yahoo.games (игры) — игры на любой вкус, как платные, так и свободно распространяемые;
- Yahoo.groups (группы) — портал представляет собой каталог сообществ по интересам;
- Yahoo.hotjobs (горячие вакансии) — сервис работодателей/соискателей, позволяющий им найти друг друга;
- Yahoo.maps (карты) — аналог сервиса google.maps, но карты несколько устарели по сравнению с google.maps, разница по сравниваемой местности где-то года в четыре;

- Yahoo.mobile (мобильные устройства) — всевозможные программы для вашего смартфона, КПК или коммуникатора;
- Yahoo.movies (кино) — все о кино, анонсы, отзывы, рекомендации, трейлеры;
- Yahoo.music (музыка) — музыка, видеоклипы, новости, анонсы;
- Yahoo.omg — актуальные фото и видео знаменитостей. Можно оставить комментарии к фотографиям;
- Yahoo.personals — сайт знакомств от Yahoo!. Огромная база пользователей из множества стран;
- Yahoo.real estate (недвижимость) — сайт, посвященный недвижимости. Можно найти объект для покупки/продажи недвижимости, проследить изменение цен и тенденций на рынке недвижимости;
- Yahoo.shopping (покупки) — каталог товаров, позволяющий не рыться в Интернете в поисках товара, а найти его тут и немедленно перейти на сайт продавца;
- Yahoo.sports (спорт) — все о спорте. Результаты игр, турнирные таблицы, лучшие игроки, самые последние новости;
- Yahoo.travel (путешествия) — подборка информации обо всех возможных маршрутах и направлениях для путешествий, стоимости проживания в местных гостиницах;
- Yahoo.TV (ТВ) — программы передач, анонсы, премьеры, новости индустрии и видеоматериалы;
- Yahoo.yellow pages (Желтые страницы) — справочник адресов и телефонов предприятий, фирм и частных предпринимателей;
- Yahoo.buzz (аналог dig.com) — любой желающий может добавить свою новость или ссылку на новость из Сети. Можно проголосовать за новость других участников и тем самым способствовать появлению этой новости на главной странице;
- Yahoo.shine — портал обо всем.

Из перечисленных функций системы видно, что они охватывают множество сфер жизнедеятельности людей.

Поисковый запрос в системах должен быть сформулирован пользователем в соответствии с тем, что он хочет найти, максимально кратко и просто. А поисковая система доставляет людям именно ту информацию, которую они ищут. Научить пользователей делать «правильные» запросы к системе, т. е. запросы, соответствующие принципам работы поисковых систем, невоз-

можно. Поэтому разработчики создают такие алгоритмы и принципы работы поисковых систем, которые бы позволяли находить пользователям искомую ими информацию. Это означает, что поисковая система «думает» так же, как думает пользователь при поиске информации.

Автоматизированные информационные системы оказывают влияние на многие характеристики работы организаций, они повышают:

- производительность труда, т. е. скорость и качество выполнения различных задач;
- функциональную эффективность, т. е. эффективность выполнения различных функций;
- качество обслуживания клиентов;
- создание и улучшение продукции;
- качество распределения продукции, закрепление клиентов и др.

В последнее время повысилась роль персональных ЭВМ, призванных существенно увеличить производительность личного труда непосредственно в процессе производства и управления. Преимущества создаются также за счет телекоммуникаций, локальных, глобальных, корпоративных сетей. С помощью информационных технологий переключение компьютера с построения диаграммы на разработку стандартного контракта или с разработки архитектурного проекта на создание карты погоды земного шара осуществляется простым изменением последовательности команд, управляющих его работой.

Взаимодействие персональных ЭВМ с сетями малых высокопроизводительных ЭВМ позволяет обеспечить потребителя необходимой информацией для управления с рабочего места, из дома, транспорта и т. д. Сегодня уже имеются готовые инструментальные программные средства, которые позволяют разрабатывать собственные проблемно-ориентированные продукты — пакеты прикладных программ. Для этого необходимо прежде всего быть хорошим специалистом в своей области, а также владеть навыками программирования.

Специалисты должны хорошо знать требования стандартов, сформированных по предметным областям в Общероссийском классификаторе стандартов. Например:

01.140 Информатика. Издательское дело

35 Информационные технологии. Машины конторские

35.020 Информационные технологии

35.040 Наборы знаков и кодирование информации
35.060 Языки, используемые в информационных технологиях
35.080 Программное обеспечение
35.110 Организация сети
35.140 Компьютерная графика
35.160 Микропроцессорные системы
35.180 Информационно-технологические терминалы и другие периферийные устройства...
35.200 Интерфейсы и межсоединительные устройства
35.220 Запоминающие устройства
35.240 Применение информационных технологий
25.040 Промышленные автоматизированные системы и др.

Любая информационная система подразумевает участие в ее работе людей — персонала, имеющего отношение к информационным системам. Выделяют такие категории персонала, как конечные пользователи, программисты, системные аналитики, администраторы баз данных и др. Программистом традиционно называют человека, который составляет программы. Человека, использующего результат работы компьютерной программы, называют конечным пользователем. Системный аналитик — это специалист, оценивающий потребности пользователей в применении компьютера, а также проектирующий информационные системы, которые соответствуют этим потребностям.

В сфере экономического менеджмента с информационными системами работают две категории специалистов: управляющие (конечные пользователи) и специалисты по обработке данных. Специалисты по обработке данных профессионально анализируют, проектируют и разрабатывают систему. Выпускники высших и средних профессиональных учебных заведений не могут рассчитывать на хорошую работу, если они не имеют навыков работы на компьютере, не владеют информационными технологиями, не обладают знаниями об АИС и АСУ.

Книга поможет студентам в приобретении таких знаний, ознакомит студентов с теоретическими и практическими основами информатики, информационных автоматизированных технологий и систем.

В первой части книги представлены темы, посвященные истории развития вычислительной техники и становлению информатики, вопросам представления, измерения и хранения информации, системам счисления, логическим основам, архитектуре и устройствам ЭВМ, основным понятиям операционных систем и

их файловой структуре, системному и прикладному программному обеспечению. Особое внимание уделено основам знаний по алгоритмизации, технологии программирования, языкам программирования, а также системе объектно-ориентированного программирования MS Visual Basic.

Во второй части книги темы посвящены информационным технологиям. Излагаются вопросы компьютерной обработки текстовой, числовой, графической информации, основы баз данных и знаний, систем управления базами данных (СУБД), дается представление о локальных и глобальных компьютерных сетях и знания о средствах создания веб-документов.

Третья часть книги посвящена вопросам разработки и функционирования автоматизированных информационных систем (АИС). Рассматриваются вопросы необходимости автоматизации информационных потоков, состав и структура АИС, методы и стадии их разработки, обеспечивающая и функциональные части, типы АИС, тенденции развития информационных систем.

Материал книги позволит студентам приобрести базовые знания и применять их на современном уровне в учебной и дальнейшей производственной деятельности.

Часть I
ИНФОРМАТИКА

Глава 1
ОСНОВНЫЕ ПОНЯТИЯ ИНФОРМАТИКИ

1.1. История развития информатики

Информатика — наука об общих свойствах и закономерностях информации, а также о методах ее поиска, передачи, хранения, обработки и использования в различных сферах деятельности человека. Как наука сформировалась в результате появления ЭВМ. Включает в себя теорию кодирования информации, разработку методов и языков программирования, математическую теорию процессов передачи и обработки информации.

Становление информатики как научной дисциплины относится к периоду создания, массового производства и широкого внедрения в практику быстродействующих автоматизированных электронных средств обработки информации.

В самом общем смысле под информатикой понимают фундаментальную естественную науку, изучающую процессы передачи, хранения и обработки информации.

Такое понятие информатики охватывает области, связанные с разработкой, созданием, использованием и обслуживанием систем обработки информации, включая машины, оборудование, математическое и программное обеспечение и организационные аспекты.

Главным в указанном определении, составляющим основу современного содержания информатики, является ЭВМ и ма-

шинная обработка информации. Отсюда следует, что основное содержание информатики составляют три неразрывно связанные между собой части: технические, алгоритмические и программные средства.

Понятие информации рассматривалось еще античными философами. После промышленной революции суть информации, вопросы теории информации стала рассматривать новая наука — кибернетика.

Кибернетика — наука об общих закономерностях в управлении и связи в различных системах: искусственных, биологических, социальных, — начала бурно развиваться после Второй мировой войны. В 1948 г. американский математик Норберт Винер выпустил книгу «Кибернетика, или Управление и связь в животном и машине». В ней показаны пути создания общей теории управления и связи. Развиваясь вместе с развитием ЭВМ, кибернетика со временем превращалась в общую науку о преобразовании информации. Можно сказать, что информатика вышла из кибернетики.

Под *информацией* в кибернетике понимается любая совокупность сигналов, воздействий или сведений, которые некоторой системой воспринимаются от окружающей среды (входная информация), выдаются в окружающую среду (выходная информация), а также хранятся в себе (внутренняя, внутрисистемная информация).

В нашей стране развитие кибернетики встретило идеологические препятствия, что нанесло серьезный ущерб развитию этой науки у нас. Так, в Философском словаре 1959 г. кибернетика характеризовалась как «буржуазная лженаука».

В мировой науке вскоре вслед за появлением термина «кибернетика» стало использоваться понятие Computer Science, а на рубеже 1960—1970-х гг. французы ввели термин Informatique. У нас в стране этот термин связывался с информационно-аналитической деятельностью в библиотечном деле, книгоиздании и т. п. Но сейчас, по словам академика А. П. Ершова, термин «информатика» «вводится в русский язык в новом и куда более широком значении — как название фундаментальной естественной науки, изучающей процессы передачи и обработки информации».

Информатика неразрывно связана с вычислительной техникой, компьютерными системами и сетями, так как именно компьютеры позволяют порождать, хранить и автоматически перерабатывать информацию в таких количествах, что научный под-

ход к информационным процессам становится одновременно необходимым и возможным.

Термин «информатика» еще не является общепринятым. Обратимся к истории.

В начале прошлого века появились новые концептуальные идеи, которые по-новому представляли картину мира. Речь шла об информационном факторе. На первый план стали выступать не отдельные предметы и явления, а их взаимозависимости. Это стало возможным благодаря появлению новых идей и подходов, которые объединяются единым понятием «формализм». Его основные черты:

- явное разделение содержательного и знакового аспектов изучаемого объекта;
- возможность формального преобразования знаков и знаковых систем;
- множественность интерпретаций знаков и знаковых систем.

Идеи формального преобразования знаковых систем привели к мысли об автоматизации и воплощении ее на ЭВМ.

В 1978 г. Международный конгресс по информатике дал такое определение: «Понятие информатики охватывает области, связанные с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая машины, оборудование, математическое обеспечение, организационные аспекты, а также комплекс промышленного, коммерческого, административного и социального воздействия». Как наука информатика изучает общие закономерности, свойственные информационным процессам в системах различной природы. Прикладное значение ее в том, что она охватывает почти все виды человеческой деятельности: производство, управление, науку, образование, проектные разработки, торговлю, финансовую сферу, медицину, криминалистику, охрану окружающей среды.

Информатика и кибернетика имеют много общего, но у них есть и различия. За кибернетикой сохраняется исследование общих законов движения информации в производственных системах, а информатика изучает способы и приемы переработки, передачи и использования информации. Некоторые ученые считают кибернетику составной частью информатики.

Но природа информации не имеет значения, когда разрабатываются новые носители информации, каналы связи, приемы кодирования, визуального отображения информации и многое

другое. Для разработчика СУБД, например, важны общие принципы организации и эффективность поиска данных, а не то, какие конкретные данные будут затем заложены в базу многочисленными пользователями. Эти общие закономерности есть *предмет информатики* как науки. *Объектом* приложений информатики являются самые разные области человеческой деятельности, для которых она стала источником современных технологий, называемых *«новые информационные технологии»*. Имея общие черты, они существенно различаются между собой. Реализованы информационные технологии могут быть как самостоятельно, так и в рамках автоматизированных информационных систем различных предметных областей. Примеры реализации информационных технологий приведены в разделе 4.2 и части III.

Составные части информатики:

1. *Теоретическая информатика.* Использует математические методы для изучения процессов обработки информации. Включает ряд математических разделов: математическую логику, теорию алгоритмов и автоматов, теорию информации и кодирования, теорию формальных языков и грамматик, исследование операций и др.

2. *Вычислительная техника.* Разрешает общие принципы построения вычислительных систем, принципы решения по архитектуре вычислительных систем (состав, назначение, функциональные возможности и принципы взаимодействия устройств).

3. *Программирование.* Разработка ПО (системного и прикладного).

4. *Информационные системы.* Решают вопросы анализа потоков информации в сложных системах, их оптимизации, структурирования, принципов хранения и поиска информации (Интернет, гипертекстовые поисковые системы в Сети, справочная система 09 и др.).

5. *Искусственный интеллект.* Область, в которой решаются сложнейшие проблемы; находится на пересечении с психологией, физиологией, лингвистикой и другими науками. Моделирование рассуждений, компьютерная лингвистика, машинный перевод, экспертные системы, распознавание образов и др. Общение человека с ПЭВМ должно стать похоже на межчеловеческое.

Дальнейшее развитие информатики предполагает массовое использование так называемых информационных роботов — новых систем поиска и обработки информации по Сети, в основе которых не только алгоритм прохода по Сети и поиска нужной

информации, но и элементы экспертной системы, позволяющей проанализировать искомую информацию и даже подготовить конечный результат в форме, подходящей для презентации. Такие системы помогут в подготовке любых документов — от школьного реферата или студенческого диплома до серьезного аналитического обзора.

Система мировой компьютерной коммуникации — это не только новая компьютерная технология передачи информации на любые расстояния. Это еще и особый мир, живая информационная среда, в которой с помощью средств связи, а точнее систем передачи данных, объединяются в единое целое источники, средства обработки и потребители информации. Это мир, в котором все люди, независимо от того, близко или далеко они находятся от центров обработки и хранения информации, имеют равные возможности доступа к ней.

Информационные технологии все активнее начинают вторгаться в различные сферы человеческой деятельности — в науку, производство, банковское дело, образование, здравоохранение и даже в домашний быт. Все больше людей во всем мире начинают работать с информацией, используя для этого телекоммуникации. Киберпространство и виртуальная реальность, которые постепенно входят в нашу жизнь, приобщают нас к информационным ресурсам всего человечества, расширяют наш кругозор и меняют сам образ жизни.

Одним из наиболее ярких явлений этого процесса является возникновение и бурное развитие глобальной информационной компьютерной сети Интернет. При появлении Интернета наиболее остро встала проблема поиска в нем нужных данных. Эта проблема послужила поводом к появлению поисковых машин.

Любая АИПС состоит из двух основных частей: формирователя собственной базы данных и генератора ответов на запросы пользователей. Главными показателями для первой части являются ее объем и продуманность внутренней структуры, а для второй — скорость поиска информации и удобство пользования. Функциональная гибкость запросов изначально зависит от структуры базы данных (БД): невозможно запросить что-то, что не было заложено в алгоритмы. Для просмотра и поиска в БД поисковых систем вводят индексацию внутренней структуры. Упрощенно под индексом можно понимать инструмент, помогающий существенно ускорить поиск (алгоритмы, способы обращения к большим объемам информации, способы упорядочения

и хранения данных и т. д.). Важно также, чтобы запрос был максимально конкретным. Тогда клиент получит наиболее точный результат.

Поисковые системы в Интернете обычно состоят из следующих компонентов:

- агент (поисковый робот, «паук»), который перемещается по Сети и собирает информацию;
- база данных, которая содержит всю информацию, собираемую поисковыми роботами;
- поисковый механизм, который используется как интерфейс для взаимодействия с базой данных.

Агенты — самые «интеллектуальные» из поисковых средств. Они могут просто искать, выполнять ваши транзакции, могут искать сайты специфической тематики и возвращать списки сайтов, отсортированных по их посещаемости. Агенты могут обрабатывать содержание документов, находить и индексировать страницы и другие виды ресурсов и т. д.

Поисковые роботы осуществляют общий поиск информации в Сети, сообщают о содержании найденного документа, индексируют его и извлекают итоговую информацию, просматривают заголовки, некоторые ссылки и посылают проиндексированную информацию в БД.

Роботы могут быть запрограммированы так, чтобы переходить по различным ссылкам разной глубины вложенности, выполнять индексацию и даже проверять ссылки в документе.

Средства поиска и структурирования, иногда называемые поисковыми механизмами (машинами), используются для того, чтобы помочь людям найти информацию, в которой они нуждаются. Средства поиска типа агентов, «пауков» и роботов используются для сбора информации о документах. Это специальные программы, которые занимаются поиском страниц в Сети, извлекают гипертекстовые ссылки на этих страницах и автоматически индексируют информацию, которую они находят, для построения базы данных. Каждый поисковый механизм имеет собственный набор правил, определяющих, как собирать документы. Некоторые следуют за каждой ссылкой на каждой найденной странице и затем исследуют каждую ссылку на каждой из новых страниц. Некоторые игнорируют ссылки, которые ведут к графическим и звуковым файлам, файлам мультипликации; другие игнорируют ссылки к ресурсам определенного типа баз данных; третьи просматривают прежде всего наиболее популярные страницы.

Администраторы поисковых систем могут определить, какие сайты или типы сайтов агенты должны посетить и проиндексировать. Проиндексированная информация отсылается базе данных поискового механизма. Люди могут помещать информацию прямо в индекс, заполняя особую форму для того раздела, в который они хотели бы поместить свою информацию. Эти данные передаются БД.

Для поиска информации, доступной в Интернете, пользователь посещает страницу поисковой системы и заполняет форму, детализирующую информацию, которая ему необходима. Здесь могут использоваться ключевые слова, даты и другие критерии. Критерии в форме поиска должны соответствовать критериям, используемым агентами при индексации информации, которую они нашли при перемещении по Сети. БД отыскивает предмет запроса и выводит соответствующие документы. Чтобы определить порядок, в котором список документов будет показан, БД применяет алгоритм ранжирования. В идеальном случае документы, наиболее релевантные пользовательскому запросу, будут размещены первыми в списке. Различные поисковые системы используют различные алгоритмы ранжирования, однако основные принципы определения релевантности следующие.

1. Количество слов запроса в текстовом содержимом документа (т. е. в HTML-коде).

2. Теги (управляющие символы), в которых эти слова располагаются.

3. Местоположение искомых слов в документе.

4. Удельный вес слов, относительно которых определяется релевантность, в общем количестве слов документа.

БД выводит ранжированный список документов и возвращает его человеку, сделавшему запрос. Разные поисковые механизмы также выбирают различные способы показа полученного списка. Некоторые показывают только ссылки; другие выводят ссылки с первыми несколькими предложениями, содержащимися в документе, или заголовок документа вместе со ссылкой. При щелчке по ссылке к одному из документов этот документ запрашивается у соответствующего сервера.

Рассмотрим АИС на примере поисковой системы Rambler — одной из старейших. Она содержит информацию о более чем 12 млн документов, расположенных на серверах России и стран СНГ. Rambler ежесуточно обрабатывает не менее 500 тыс. поисковых запросов (5 запросов в секунду), сканируя 48 тыс. веб-сер-

веров и используя несколько одновременно работающих программ-роботов.

Запрос может состоять из одного или нескольких слов, разделенных пробелами. Могут быть использованы как русские, так и английские слова и словосочетания. По умолчанию находятся только те документы, в которых встретились все введенные слова. Чтобы найти документы, содержащие хотя бы одно слово из запроса, используют логическую связку ИЛИ (Or) либо выбирают на странице детального запроса: **Слова запроса: любое**. Чтобы исключить документы, содержащие те или иные слова, их следует указать на странице детального запроса: **Исключить документы, содержащие следующие слова...**.

Регистр при построении запроса не имеет значения.

Слова запроса могут быть соединены логическими связками И (And), ИЛИ (Or). Вместо связок (или в сочетании с ними) могут использоваться также символы & и | соответственно. Части запроса могут быть сгруппированы с помощью круглых скобок. Возможна многократная вложенность скобок в сочетании с логическими операторами.

Rambler умеет искать слова во всех формах (например, издательство, издательства, издательством и т. д.). Чтобы слово находилось во всех формах, перед ним надо поставить служебный символ #. В меню детального запроса такой режим может быть включен для всех слов: **Расширение запроса: все формы слов**. Служебный символ @ перед словом позволяет находить не только это слово, но и однокоренные слова. В меню детального запроса символу @ соответствует режим **Расширение запроса: все однокоренные**.

Чтобы уменьшить «шум» в найденных документах, по умолчанию система ищет слова запроса так, как они были введены. Для обозначения произвольной части слова и произвольного символа или расширения запроса, можно использовать метасимволы * и ?. Ограничить поиск частями документов можно через меню детального запроса **Искать в...**. Чтобы ограничить поиск документами только на русском или только на английском языке, надо выбрать соответствующий режим в меню детального запроса **Язык документа...**. По умолчанию поиск выполняется по документам на всех языках. По умолчанию найденные документы сортируются по релевантности (соответствию запросу). Однако можно указать, чтобы вместо этого в начало списка были помещены самые свежие (или, наоборот, самые старые документы), для этого надо выбрать соответствующую установку в меню **Сор-**

тировать по... на странице детального запроса. Ограничение поиска документов определенным периодом времени достигается указанием на странице детального запроса значений **От даты ... до даты ....**

В 1993 г. А. Волок и И. Сегалович предложили технологию поиска неструктурированной информации с учетом русского языка. В 1996 г. линейку программ для этого поиска назвали Яндексом (Языковый Индекс). В 1997 г. была запущена одноименная поисковая система (www.yandex.ru), которая является одной из самых посещаемых (ежедневно около 9 млн человек). В настоящее время Яндекс — крупнейший информационный портал русскоязычного Интернета (Рунета), предоставляющий поисковый сервис, бесплатную почту, новости, каталоги товаров, платежную систему и многое другое.

Робот Яндекса обходит Сеть за несколько дней и индексирует открытую часть Сети — страницы, которые доступны при переходе по ссылке, без ввода логина и пароля. По мере роста системы основная нагрузка ложится уже не на индексирующего робота, а на поиск и выдачу результатов. Первичная индексация позволяет появиться странице уже через 2—4 ч в результатах поиска. На полную индексацию страница ставится в очередь. Качественный поиск обеспечивается тщательной лингвистической и технической проработкой текстов. Яндекс постоянно совершенствует алгоритм расчета релевантности (меры соответствия результатов поиска поставленному запросу) страниц. Поисковая машина проводит не только анализ текста, но и учитывает положение страницы на сайте, авторитетность источника, частоту обновления, цитируемость страниц. Яндекс ежедневно решает проблемы корректного ранжирования, обновления и синхронизации индекса, работы с различными языками и кодировками, дублирующейся информации, обновления базы данных и др.

Яндекс разрабатывает полезные веб-сервисы, различные программы: как бесплатные приложения для массового пользователя, так и коммерческие серверные решения.

Имеются программы:

- для поиска по веб-сайту или локальной сети с учетом морфологии русского языка;
- для обмена мгновенными сообщениями и получения уведомлений о новых письмах в Яндекс.Почта. Можно также выполнять поиск в Интернете, следить за прогнозом погоды и загруженностью дорог;

MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
38 7144
AXBOROT-RESURS MARKAZI

- для персонального поиска. Эту программу можно установить на компьютере и осуществлять с ее помощью полнотекстовый поиск с учетом морфологии русского языка по файлам разных форматов;
- mystem — производит морфологический анализ текста на русском языке. Для слов, отсутствующих в словаре, порождаются гипотезы;
- Яндекс.Бар — расширение, которое встраивается в браузер в виде дополнительной панели инструментов. Имеется много возможностей — от удобного и быстрого поиска до сохранения закладок на серверах Яндекса и уведомлений о свежих сообщениях в почте и лент новостей;
- спамообороны — серверное решение для фильтрации спама, предназначенное для корпоративных пользователей и интернет-провайдеров. Продукт основан на технологии, уже несколько лет применяемой для защиты пользователей. Система выполняет комплексный анализ тысяч характеристик письма с учетом их значимости, обеспечивая высокую полноту и точность фильтрации.

По данным Яндекса, на сентябрь 2006 г. в Рунете имелось более 1 150 000 блогов. Ежемесячно поиском по блогам Яндекса пользовались около полумиллиона человек. В 2005 г. каждый час появлялось 20 новых блогов, в 2006 г. — уже более 100. Большинство авторов дневников (более 60 %) ведут их на российских блог-хостингах. Сегодня русскоязычные блоггеры ведут дневники на местных сервисах почти в два раза охотнее, чем на западных. Каждый день на российских блог-хостингах появляется около 1400 новых блогов. В блогосфере находят свое отражение события культурной, политической, спортивной, международной жизни и др. Поиск по блогам Яндекса — инструмент для «навигации по общественному мнению» в Интернете.

В 1998 г. сотрудники Стенфордского университета (США) Сергей Брин и Лоуренс Пейдж создали ИПС Google. Система имеет более двух миллиардов проиндексированных страниц. Google, в отличие от других иностранных ИПС, хорошо индексирует русскоязычные веб-ресурсы. Внутренняя система оценки качества документов, называемая PageRank (PR), при решении о порядке выдачи пользователю списка страниц, попадающих под его запрос, во внимание принимает некий коэффициент, зависящий от количества ссылок с других сайтов на эту страницу и от их популярности. PageRank — это нормализованное отношение количе-

ства ссылок, приводящих на данную страницу, к количеству исходящих с нее. Имеется расчетная формула. См. также раздел 10.1.

Качество поисковой системы характеризуется не только количеством проиндексированных документов, правилами их отбора в итоговый список, но и тем, как часто интернет-роботы заново проверяют содержимое ранее обработанных сайтов.

На рис. 1.1 (пример из thermo.karelia.ru) приведена таблица данных об общем объеме запрошенных роботами документов и количестве посещений сайта (по данным Webalizer — анализатора журналов веб-сервера).

ИПС	Май	Июнь	Июль	Август	Сентябрь	Всего за 5 месяцев
«Яндекс»	16 (5)	8 (1)	21 (2)	10 (1)	48 (6)	103 (15)
«Апорт»	—	18 (4)	21 (2)	28 (3)	—	72 (11)
«Рамблер»	—	11 (3)	—	10 (7)	8 (5)	29 (15)
FAST	17 (10)	—	—	16 (8)	23 (13)	56 (31)
Google	—	—	—	35 (127)	50 (204)	85 (331)
Altavista	—	—	—	—	18 (28)	18 (28)
sunet.ru	33 (8)	34 (8)	34 (8)	45 (10)	47 (5)	193 (39)
dec.com	—	24 (21)	19 (26)	38 (39)	16 (13)	97 (99)

Рис. 1.1. Объем документов, загруженных роботами ИПС, и количество посещений сайта за месяц

Видно, что роботы Яндекса и Google ведут себя по-разному. Яндекс останавливается на корневом документе веб-сервера и скачивает содержимое сайта последовательно, документ за документом, в один поток. Google распараллеливает работу между несколькими роботами, причем каждый из них при скачивании может «отвлекаться» на другие дела. То есть эти две поисковые системы характеризуются совершенно различными структурами и способами пополнения информации из Интернета.

Второй важный вывод, который напрашивается по результатам изучения приведенной таблицы, заключается в том, что русскоязычные поисковики посещают ресурсы постсоветского пространства чаще иностранных. Даже Апорт, уступивший на

данный момент третье место Google по общему количеству обрабатываемых запросов, как минимум раз в месяц просматривает содержимое каждого сайта.

Dec.com — робот некогда существовавшего гиганта Digital Equipment Corporation (впоследствии он был перекуплен компанией Compaq). Dec поддерживал AltaVista, поэтому можно предположить, что получаемая информация идет в закрома самой известной ИПС.

Заметим, что каждый сайт строится уникальным образом, имея линейную, древовидную или смешанную структуру. Одни построены на документах с динамическим контентом, другие вообще статичны и не изменяются. По-видимому, будет существовать разница даже в процессе индексации постоянно изменяющихся сайтов, имеющих динамическое содержимое. Не секрет, что целая категория веб-ресурсов, отнесенных к разделу новостных, просматривается чуть ли не ежеминутно.

1.2. Основные понятия и определения

В окружающем нас материальном мире мы постоянно сталкиваемся с *физическими телами* либо *физическими полями*. Физические объекты находятся в непрерывном движении и изменении, что сопровождается обменом энергией и ее переходом из одной формы в другую. Все виды *энергообмена* сопровождаются появлением *сигналов*, т. е. все сигналы имеют материальную энергетическую природу. При взаимодействии сигналов с физическими телами в последних возникают определенные изменения свойств — это явление называется *регистрацией сигналов*. Сообщение, отображающее информацию, всегда представляется в виде сигнала (изменения состояния некоторого объекта). В зависимости от физической среды сигналы могут быть механические, электрические, световые, звуковые и др. Они могут быть статические и динамические, непрерывные и дискретные. Изменения можно наблюдать, измерять или фиксировать разными способами — при этом возникают и регистрируются новые сигналы, т. е. образуются данные.

Данные — это зарегистрированные сигналы. Данные — информация, представленная в виде, позволяющем хранить, передавать ее или обрабатывать как человеком, так и в технических средствах.

Кодирование данных — перевод данных из одной системы обозначений в другую, как правило, более оптимальную. Например, перевод команд алгоритма на команды строго формализованного языка, языка программирования.

Системы счисления — совокупность приемов наименования и записи чисел.

Информация — это продукт взаимодействия данных и адекватных им методов, сведения (разъяснение, изложение, осведомленность), полученные в результате этого взаимодействия.

Знания — проверенный практикой или доказанный теоретически результат познания действительности, вернее, ее отображения в сознании человека. Знание — категория, отражающая передачу накопленного человеческого опыта от предков к потомкам.

Информатика (в США: Computer Science — компьютерная наука; в Великобритании: Computing Science — вычислительная наука) — наука о способах получения, накопления, хранения, преобразования, передачи и использования информации. На современном этапе обязательно с использованием компьютеров и телекоммуникаций.

Информационный процесс — процесс получения, создания, сбора, обработки, накопления, хранения, поиска, распространения и использования информации.

Информационная технология — процесс, использующий совокупность средств и методов обработки и передачи первичной информации для получения информации нового качества о состоянии объекта, процесса или явления.

Информационная система — взаимосвязанная совокупность средств, методов и персонала, участвующих в обработке данных.

1.3. Информация и информационные процессы

Данные несут в себе информацию о событиях, произошедших в материальном мире, поскольку они являются регистрацией сигналов, возникших в результате этих событий. Однако данные не тождественны информации. Наблюдая излучения далеких звезд, человек получает определенный поток данных, но станут ли эти данные информацией, зависит еще от очень мно-

гих обстоятельств. Для того чтобы данные стали информацией, необходимо наличие метода пересчета одной физической величины в другую, т. е. метода обработки данных. Таким образом, *информация* — это продукт взаимодействия данных и адекватных им методов. Появление информации является естественным следствием развития человеческого общества. Термин «информация» происходит от латинского слова information — разъяснение, изложение, осведомленность. То есть это некоторые сведения, совокупность полученных знаний. Понятие информации должно быть связано с определенным объектом, свойства которого она отображает. Однако наблюдается некоторая независимость информации от ее носителя.

Информация является динамическим объектом, она существует только в момент протекания информационного процесса, в остальное время она находится в виде данных.

В определении Н. Винера, кибернетика — наука управления и изучения информационных процессов, т. е. она изучает общие закономерности передачи информации. *Информационный процесс* — это функционирование всех элементов информационной среды и всех факторов, обеспечивающих появление новых знаний, их передачу, переработку, использование и воздействие на объект рассматриваемой системы.

Под информационным процессом понимают процесс взаимодействия между двумя объектами материального мира, в результате которого возникает информация, являющаяся результатом отображения одного объекта другим. В ходе информационного процесса данные преобразуются из одного вида в другой с помощью методов.

Обработка данных включает в себя множество различных операций: сбор, формализацию, фильтрацию, сортировку, архивацию, защиту, транспортировку, преобразование и др.

Миллионы людей во всем мире занимаются созданием, обработкой, преобразованием и транспортировкой данных, и на каждом рабочем месте выполняются свои специфические операции, необходимые для управления социальными, экономическими, промышленными, научными и культурными процессами. Эта работа с информацией может иметь огромную трудоемкость, и ее надо автоматизировать.

Для автоматизированной обработки данные должны быть упорядочены, т. е. структурированы, иметь определенную структуру. Существует три основных типа структур данных: линейная,

иерархическая, табличная. Например, если собрать все листы книги в правильной последовательности, мы получим простейшую структуру данных — *линейную*. Такую книгу можно читать, хотя для поиска нужных данных все листы придется прочитать подряд, начиная с самого начала, что не всегда удобно.

Линейные структуры — это списки. *Список* — это простейшая структура данных, отличающаяся тем, что каждый элемент данных однозначно определяется своим номером в массиве. Проставляя номера на отдельных страницах книги, мы создаем структуру списка.

Для быстрого поиска данных существует *иерархическая структура*. Например, книгу разбивают на части, разделы, главы, параграфы и т. п. Элементы структуры более низкого уровня входят в элементы структуры более высокого уровня: разделы состоят из глав, главы из параграфов и т. д. Для больших массивов поиск данных в иерархической структуре намного проще, чем в линейной, однако и здесь требуется *навигация*, связанная с необходимостью просмотра.

Задачу упрощают тем, что в большинстве книг есть вспомогательная перекрестная *таблица*, связывающая элементы иерархической структуры с элементами линейной структуры, т. е. связывающая разделы, главы и параграфы с номерами страниц. В книгах с простой иерархической структурой, рассчитанных на последовательное чтение, эту таблицу принято называть *оглавлением*, а в книгах со сложной структурой, допускающей выборочное чтение, ее называют *содержанием*.

1.4. Кодирование данных

Для автоматизации работы с данными очень важно унифицировать их форму представления. Для этого используется прием *кодирования*, т. е. выражение данных одного типа через данные другого типа. Естественные человеческие *языки* — это не что иное, как системы кодирования понятий для выражения мыслей посредством речи. К языкам близко примыкают *азбуки* (системы кодирования компонентов языка с помощью графических символов). Проблема универсального средства кодирования достаточно успешно реализуется в отдельных отраслях техники, науки и культуры. В качестве примеров можно привести систему записи математических выражений, телеграфную азбу-

ку, морскую флажковую азбуку, систему Брайля для слепых и многое другое. Своя система существует и в вычислительной технике. Это *двоичное кодирование*, основанное на представлении данных последовательностью двух знаков: 0 и 1, которые называются *двоичными цифрами*, по-английски — binary digit, или, сокращенно, *bit* (*бит*).

Одним битом могут быть выражены два понятия (да или нет, черное или белое, истина или ложь и т. п.). Двумя битами можно выразить уже четыре понятия: 00 01 10 11. Тремя битами можно закодировать восемь значений:

000 001 010 011 100 101 110 111.

Увеличивая на единицу количество разрядов в системе двоичного кодирования, мы увеличиваем в два раза количество значений, которое может быть выражено в данной системе, т. е. общая формула имеет следующий вид:

$$N = 2^m,$$

где N — количество независимых кодируемых значений; m — разрядность двоичного кодирования, принятая в данной системе.

Кодирование целых и действительных чисел. Целые числа кодируются двоичным кодом довольно просто — достаточно взять целое число и делить его пополам до тех пор, пока в остатке не образуется нуль или единица. Совокупность остатков от каждого деления, записанная справа налево вместе с последним остатком, и образует двоичный аналог десятичного числа.

$$19 : 2 = 9 + 1; \quad 9 : 2 = 4 + 1; \quad 4 : 2 = 2 + 0; \; 2 : 2 = 1.$$

Таким образом, $19_{10} = 1011_2$.

Для кодирования целых чисел от 0 до 255 достаточно иметь 8 разрядов двоичного кода (8 бит). 16 бит позволяют закодировать целые числа от 0 до 65 535, а 24 бит — уже более 16,5 млн разных значений.

Для кодирования действительных чисел используют 80-разрядное кодирование. При этом число предварительно преобразуется в *нормализованную форму*:

$3{,}1415926 = 0{,}31415926 \cdot 10^1$;

$300\,000 = 0{,}3 \cdot 10^6$;

$123\,456\,789 = 0{,}123456789 \cdot 10^{10}$.

Первая часть числа называется *мантиссой*, а вторая — *характеристикой*. Большую часть из 80 бит отводят для хранения мантиссы (вместе со знаком), и некоторое фиксированное количество разрядов отводят для хранения характеристики (тоже со знаком).

Кодирование текстовых данных. Если каждому символу алфавита сопоставить определенное целое число, то с помощью двоичного кода можно кодировать текстовую информацию. Восьми двоичных разрядов достаточно для кодирования 256 различных символов. Этого хватит, чтобы выразить различными комбинациями восьми битов все символы английского и русского языков, как строчные, так и прописные, а также знаки препинания, символы основных арифметических действий и некоторые общепринятые специальные символы, например символ §.

Институт стандартизации США (ANSI — American National Standard Institute) ввел в действие систему кодирования ASCII (American Standard Code for Information Interchange — американский стандартный код обмена информацией). В ASCII закреплены две таблицы кодирования — базовая и расширенная. Базовая таблица закрепляет значения кодов от 0 до 127, а расширенная — от 128 до 255.

0—32 — коды базовой таблицы для производителей аппаратных средств; 32—127 — коды символов английского алфавита, знаков препинания, цифр, арифметических действий и некоторых вспомогательных символов.

Пример системы кодирования — код ASCII:

```
43 4F 4D 50 55 54 45 52.
```

Имеются и другие кодировки: Windows 1251 (коды 128—255), КОИ-8 (128—255), ISO (160—255), ГОСТ — альтернативная кодировка (128—255). Универсальная система, основанная на 16-разрядном кодировании символов (а не 8-разрядном), Unicode, позволяет обеспечить коды для 65 536 различных символов. Этого достаточно для всех языков планеты.

Кодирование графических данных. Черно-белое графическое изображение состоит из мельчайших точек, образуюших характерный узор, называемый *растром*. Растр — это метод кодирования графической информации, издавна принятый в полиграфии. Линейные координаты и индивидуальные свойства каждой точки (яркость) можно выразить с помощью целых чисел.

Растровое кодирование позволяет использовать двоичный код для представления графических данных. Общепринятым считается представление черно-белых иллюстраций в виде комбинации точек с 256 градациями серого цвета, и, таким образом, для кодирования яркости любой точки обычно достаточно восьмиразрядного двоичного числа. Для кодирования цветных графических изображений применяется *принцип декомпозиции* произвольного цвета на основные составляющие. Используют три основных цвета: красный (Red, R), зеленый (Green, G) и синий (Blue, B).

На практике считается (хотя теоретически это не совсем так), что любой цвет, видимый человеческим глазом, можно получить путем механического смешения этих основных цветов. Такая система кодирования называется системой RGB. Если для кодирования яркости каждой из основных составляющих использовать по 256 значений (восемь двоичных разрядов), как это принято для полутоновых черно-белых изображений, то на кодирование цвета одной точки надо затратить 24 разряда. При этом система кодирования обеспечивает однозначное определение 16,5 млн различных цветов, что на самом деле близко к чувствительности человеческого глаза. Такой режим представления цветной графики называется *полноцветным* (*True Color*). Каждому из основных цветов можно поставить в соответствие дополнительный, т. е. цвет, дополняющий основной цвет до белого. Для любого из основных цветов дополнительным будет цвет, образованный суммой пары остальных основных цветов. Дополнительными цветами являются: голубой (Cyan, C), пурпурный (Magenta, M) и желтый (Yellow, Y). Любой цвет можно представить в виде суммы голубой, пурпурной и желтой составляющих. Такой метод кодирования цвета принят в полиграфии, но в полиграфии используется четвертая краска — черная (Black, K). Эта система кодирования обозначается четырьмя буквами *CMYK* (черный цвет обозначается буквой K). Для представления цветной графики в этой системе надо иметь 32 двоичных разряда. Такой режим тоже называется полноцветным. Кодирование цветной графики 16-разрядными двоичными числами называется режимом *высококачественного цвета* (*High Color*). При кодировании информации о цвете с помощью 8 бит данных можно передать только 256 цветовых оттенков. Такой метод кодирования цвета называется *индексным*. 256 значений недостаточно, чтобы передать весь диапазон цветов, поэтому код каждой точки

растра выражает не цвет, а только его номер (индекс) в справочной таблице, называемой *палитрой*.

Кодирование звуковой информации. Метод *FM* (*Frequency Modulation*) основан на том, что теоретически любой сложный звук можно разложить на последовательность простейших гармонических сигналов разных частот, каждый из которых представляет собой правильную синусоиду и может быть описан числовыми параметрами, т. е. кодом. В природе звуковые сигналы имеют непрерывный спектр, т. е. являются аналоговыми. Их разложение в гармонические ряды и представление в виде дискретных цифровых сигналов выполняют специальные устройства — *аналогово-цифровые преобразователи* (*АЦП*). Обратное преобразование для воспроизведения звука, закодированного числовым кодом, выполняют *цифро-аналоговые преобразователи* (*ЦАП*). При преобразованиях неизбежны потери информации, связанные с методом кодирования, поэтому качество звукозаписи обычно получается не вполне удовлетворительным и соответствует качеству звучания простейших электромузыкальных инструментов с окрасом, характерным для электронной музыки. Но этот метод обеспечивает весьма компактный код.

Метод *таблично-волнового* (*Wave-Table*) синтеза соответствует современному уровню развития техники. Упрощенно можно сказать, что где-то в заранее подготовленных таблицах хранятся образцы звуков для множества различных музыкальных инструментов. В технике такие образцы называют *сэмплами*. Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения, некоторые параметры среды, в которой происходит звучание, а также прочие параметры. Поскольку в качестве образцов используются реальные звуки, то качество звука, полученного в результате синтеза, получается очень высоким и приближается к качеству звучания реальных музыкальных инструментов.

1.5. Единицы представления, измерения и хранения информации

Существует два подхода в определении понятия «количество информации». В 1948 г. американский математик Клод Шеннон развил *вероятностный подход* к измерению количества информации, а работы по созданию ЭВМ привели к *объемному подходу*.

1.5.1. Вероятностный подход измерения количества информации

Пример 1.1. Бросаем правильную игральную кость из N граней (например, $N = 6$). Выпасть грани могут с одним из следующих знаков: 1, 2, ..., N. Введем в рассмотрение численную величину, измеряющую неопределенность — энтропию H. N и H связаны некоторой функциональной зависимостью:

$$H = f(N), \tag{1}$$

а сама функция f является возрастающей, неотрицательной и определенной ($N = 1, 2, ..., 6$).

Более подробно:

1) готовимся бросить кость — существует неопределенность $H1$;

2) кость брошена, информация получена, обозначим ее J;

3) неопределенность опыта после его осуществления $H2$.

За количество информации, которая получена в ходе осуществления опыта, примем разность неопределенностей до и после опыта:

$$J = H1 - H2. \tag{2}$$

Если есть результат, неопределенность снята ($H2 = 0$) и поэтому количество полученной информации совпадает с первоначальной энтропией, т. е. неопределенность, заключенная в опыте, совпадает с информацией об исходе этого опыта.

Определение вида функции f. Если варьировать число граней N и число бросаний M, то общее число исходов X (векторов длины M из знаков 1, 2, ..., N) будет:

$$X = N^M. \tag{3}$$

Пример 1.2. Бросим два раза кость с шестью гранями: $X = 6^2 = 36$.

Каждый исход X есть некоторая пара $X1$, $X2$, где $X1$, $X2$ — исходы первого и второго бросаний (общее число таких пар — X).

Ситуацию с бросанием кости M раз можно рассматривать как некую сложную систему, состоящую из независимых подсистем — однократных бросаний кости. Энтропия такой систе-

мы в M раз больше, чем энтропия одной системы (так называемый принцип аддитивности энтропии).

$$f(6^M) = M \cdot f(6),$$

т. е.

$$f(N^M) = M \cdot f(N). \tag{4}$$

Прологарифмируем левую и правую части формулы (3):

$$\ln X = M \cdot \ln N; \quad M = \ln X / \ln N,$$

подставим M в (4):

$$f(X) = \ln X / \ln N \cdot f(N).$$

Обозначив через K положительную константу (обычно принимают $K = 1/\ln 2$), получим:

$$f(X) = K \cdot \ln X.$$

Учитывая формулу (1), запишем:

$$H = K \cdot \ln N$$

или

$$H = \log_2 \cdot N \text{ — } \textit{формула Хартли.} \tag{5}$$

Что принять за единицу измерения. Если $N = 2$, то $H = 1$. В качестве единицы измерения количества информации принимается количество информации, связанное с проведением опыта, состоящего в получении одного из двух равновероятных исходов (орел, решка). Такая единица количества информации называется битом.

Все N исходов равновероятны, поэтому на долю каждого приходится $1/N$-я часть общей неопределенности опыта: $(\log_2 \cdot N)/N$. При этом вероятность i-го исхода $P_i = 1/N$. Таким образом,

$$H = \sum_{i=1}^{N} P_i \log_2 (1/P_i) \text{ — } \textit{формула Шеннона.} \tag{6}$$

Формула верна, когда Pi (вероятности i-го исхода) могут быть различны, т. е. неравновероятны.

Пример 1.3.

1. Определить количество информации, связанное с появлением каждого символа в сообщениях, записанных на русском языке. Алфавит состоит из 33 букв и знака пробела для разделения слов. По формуле (5) $H = \log_2 34 = 5$ бит ($2^5 = 32$) — максимальное количество информации, которое могло бы приходиться на один знак. Однако разные буквы встречаются с различной частотой. Тогда, используя вероятность (специальные таблицы), по формуле (6) получим $H = 4{,}72$ бит.

2. Для алфавитов из 26 букв и пробела (английский, немецкий, французский) $H = \log_2 27 = 4{,}76$ бит.

3. Алфавит состоит из двух знаков: 0 и 1. Если вероятности их появления одинаковы, ($P(0) = P(1) = 0{,}5$), то количество информации на один знак при двоичном кодировании равно $H = \log_2 2 = 1$ бит ($2^1 = 2$).

Таким образом, количество информации (в битах), заключенное в двоичном слове, равно числу двоичных знаков в нем.

К. Шеннон обобщил случай, когда H зависит не только от M, но и от вероятностей возможных выборов $H_i = \log(1/P_i) = -\log P_i$ — вероятность выбора i-го символа алфавита, $H = -\sum_{j=1}^{M} P_j \log P_j$ — среднее значение количества информации на 1 символ алфавита, а $j = 1$, M. $N = 2_J$ — количество возможных событий.

Например, если $J = 4$ бита, то $N = 2^4 = 16$.

За *единицу количества информации* принято такое количество информации, которое содержит сообщение, уменьшающее неопределенность знания в два раза. Такая единица называется битом.

1.5.2. Объемный подход измерения количества информации

В технических устройствах двоичной системе счисления отдают предпочтение, так как наиболее просто реализовать два противоположных физических состояния: заряжен/не заряжен; намагничен/не намагничен, в двух противоположных направлениях и т. д. Объем информации, записанный двоичными знаками в памяти компьютера или на внешнем носителе, подсчитыва-

ется просто по количеству требуемых для такой записи двоичных символов. Нецелое число невозможно. Для удобства введены следующие понятия:

- двоичное слово — байт информации, из 8 бит;
- 1024 байт — килобайт (Кб);
- 1024 Кб — мегабайт (Мб);
- 1024 Мб — гигабайт (Гб).

Между вероятностным и объемным подходами соотношение неоднозначное. Не всегда текст, записанный двоичными символами, допускает его измерение в кибернетическом смысле, но всегда допускает в объемном. Измерения не всегда совпадают. Кибернетическое количество не может быть больше объемного. Чаще используется объемное измерение информации.

1.6. Системы счисления

Арифметические основы ЭВМ строятся на системах счисления.

Системой счисления называется совокупность приемов наименования и записи чисел. В любой системе выбирают базисные числа (слова или знаки), а остальные получают в результате различных операций. Системы счисления различаются выбором базисных символов и правилами образования из них других чисел. Например:

- арабская система цифр: 0, 1, 2, ..., 9 (базисные числа);
- римская система счисления: I, V, X, L, C, D, M — соответственно 1, 5, 10, 50, 100, 500, 1000 (базисные символы). Система неудобна для операций умножения и деления. Правило: если цифра справа больше цифры слева или равна ей, то цифры складываются, а если цифра справа меньше цифры слева или равна ей, то левая вычитается из правой. Например, число 146 имеет вид CXLVI.

Системы счисления, в которых любое число получается путем сложения или вычитания базисных чисел, называются аддитивными. Римская система счисления — *непозиционная*, так как значение числового знака не зависит от расположения в записи числа. Сейчас в основном используют *позиционные* системы счисления, в которых значение цифры зависит от ее положения в числе.

В *десятичной* позиционной системе счисления десять единиц каждого разряда объединяются в одну единицу соседнего стар-

шего разряда. Каждый разряд имеет вес, равный степени 10. Например, 525,15 — самая левая цифра 5 означает 5 сотен (вес 10^2), а самая правая 5 означает 5 сотых (вес 10^{-2}), в середине цифра 2 означает 2 десятка (вес 10^0). Таким образом, можно записать:

$$5 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0 + 1 \cdot 10^{-1} + 5 \cdot 10^{-2}.$$

В обобщенном виде эту запись представим следующим образом:

$$X = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \ldots + a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + \ldots$$
$$\ldots + a_{-m} \cdot 10^{-m}, \qquad (1)$$

где каждый коэффициент a_i может быть одним из чисел. K единиц какого-либо разряда, объединенных в единицу более старшего разряда, называются *основанием позиционной системы счисления*, а сама система называется *K-ичной*. 10 — основание десятичной системы, 3 — основание троичной системы и т. д. В K-ичной системе

$$X = a_n \cdot K^n + a_{n-1} \cdot K^{n-1} + \ldots$$
$$\ldots + a_1 \cdot K^1 + a_0 \cdot K^0 + a_{-1} \cdot K^{-1} + \ldots + a_{-m} \cdot K^{-m}. \qquad (2)$$

Арифметические действия над числами в любой позиционной системе счисления производятся по тем же правилам, что и в десятичной системе, так как все они основываются на правилах выполнения действий над полиномами. Учитывать надо основание K системы. Основание в системе счисления определяет перенос в старший разряд числа. Условно оно указывается индексом внизу числа.

Двоичная система счисления — система с наименее возможным основанием, используется только две цифры. Запишем полином:

$$X = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \ldots + a_1 \cdot 2^1 + a_0 \cdot 2^0 +$$
$$+ a_{-1} \cdot 2^{-1} + \ldots + a_{-m} \cdot 2^{-m}, \qquad (3)$$

где каждый коэффициент a_i может быть либо 0, либо 1. Примеры записи чисел в двоичной системе счисления:

$1 = 1_2$; $2 = 10_2$; $3 = 11_2$; $4 = 100_2$; $5 = 101_2$; $6 = 110_2$;

$7 = 111_2$; $8 = 1000_2$; $9 = 1001_2$; $10 = 1010_2$;
$0{,}5 = 0{,}1_2$; $0{,}25 = 0{,}01_2$.

Пример 1.4. $A_2 = 101{,}01$. $A_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}$.

Умножение или деление двоичного числа на 2 приводит к перемещению запятой:

$101{,}01_2 \cdot 2 = 1010{,}1_2$.

Неудобство двоичной системы в том, что запись чисел громоздка. Это не имеет существенного значения для ЭВМ. Но если необходимо кодировать информацию вручную (например, при составлении программы на машинном языке), то лучше пользоваться восьмеричной или шестнадцатеричной системой счисления. Характеристики некоторых систем счисления представлены в табл. 1.1.

Таблица 1.1. **Характеристики наиболее употребимых систем счисления**

Система счисления	Основание	Алфавит цифр
Десятичная	10	0, 1, 2, ..., 9
Двоичная	2	0, 1
Восьмеричная	8	0, 1, 2, ..., 7
Шестнадцатеричная	16	0, 1, 2, ..., 9, A (10), B (11), C (12), D (13), E (14), F (15)

Восьмеричная система счисления:

$$A_8 = 673{,}2_8;\ A_8 = 6 \cdot 8^2 + 7 \cdot 8^1 + 3 \cdot 8^0 + 2 \cdot 8^{-1}.$$

Шестнадцатеричная система счисления:

$$A_{16} = 8A\ F_{16};\ A_{16} = 8 \cdot 16^1 + 10 \cdot 16^0 + 15 \cdot 16^{-1}.$$

1.6.1. Перевод чисел из одной системы счисления в другую

Исходные и конечные результаты нужно получить, как правило, в десятичной системе счисления. Для ЭВМ необходимо перевести числа из десятичной системы счисления в двоичную и наоборот.

Возьмем целые числа от 0 до $k - 1$, где k — основание системы счисления. Задача перевода заключается в следующем. Пусть известна запись числа X с основанием k:

$k_n, k_{n-1}, \ldots, k_1, k_0, k_{-1}, \ldots, k_{-2}, \ldots$ — цифры p-ичной системы ($0 \le k_i \le k_{-1}$). Требуется найти запись этого числа в системе счисления с другим основанием — q. Искомые цифры будут:

$$q_n, q_{n-1}, \ldots, q_1, q_0, q_{-1}, \ldots, q_{-2}, \ldots (0 \le q_i \le q_{-1}).$$

Перевод чисел из двоичной системы счисления в десятичную:

$$10{,}11_2 = 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} =$$

$$= 2 + 0 + 1/2 + 1/4 = 2{,}75_{10}.$$

Перевод чисел из восьмеричной системы счисления в десятичную:

$$372{,}5_8 = 3 \cdot 8^2 + 7 \cdot 8^1 + 2 \cdot 8^0 + 5 \cdot 8^{-1} =$$

$$= 3 \cdot 64 + 56 + 2 \cdot 1 + 5 \cdot 1/8 = 250{,}125_{10}.$$

Перевод чисел из шестнадцатеричной системы счисления в десятичную:

$$19F_{16} = 1 \cdot 16^2 + 9 \cdot 16^1 + 15 \cdot 16^0 =$$

$$= 1 \cdot 256 + 144 + 15 = 415_{10}.$$

Перевод чисел из десятичной системы счисления в другие. Разложим целое число в ряд по основанию 2.

$$A = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \ldots + a_1 2^1 + a_0 2^0.$$

Разделим A на основание 2.

$$A/2 = a_n \cdot 2^{n-1} + a_{n-1} \cdot 2^{n-2} + \ldots + a_1.$$

Остаток равен a_0. Далее частное опять разделим на 2. Остаток теперь равен a_1.

Тогда после n-го деления получим последовательность остатков $a_0, a_1, \ldots, a_n$.

Их последовательность совпадает с обратной последовательностью цифр целого двоичного числа в свернутой форме:

$$A_2 = a_n a_{n-1} \ldots a_1 a_0.$$

Таким образом, достаточно записать остатки в обратной последовательности, чтобы получить искомое двоичное число. Шаги:

1. Делить десятичное число и остаток на 2 до тех пор, пока не получим частное меньше делителя 2.
2. Записать остатки в обратной последовательности, чтобы получить искомое число в двоичной системе счисления.

Пример 1.5. Перевести число 19 в двоичную систему счисления. 19/2, остаток равен $1 = a_2$; далее результат 9/2, остаток равен $1 = a_1$; далее результат 4/2, остаток равен $0 = a_2$; далее 2/2, остаток равен $0 = a_3$, далее результат деления 1/2 — в остатке $1 = a_4$, т. е. $A_2 = a_4 a_3 a_2 a_1 a_0 = 10011_2$.

Перевод чисел из двоичной системы счисления в восьмеричную (основание — $2^n = 2^3$). Каждый разряд двоичного числа содержит 1 бит. Для записи восьмеричных чисел используется 8 цифр, т. е. в каждом разряде числа возможны 8 вариантов записи. Решаем показательное уравнение $8 = 2^J$. Так как $8 = 2^3$, то $J = 3$ бита, т. е. каждый разряд восьмеричного числа содержит 3 бита — триада. Для перевода двоичного числа в восьмеричное надо разбить его на группы по 3 цифры справа налево, а затем преобразовать каждую группу в восьмеричную цифру (последнюю левую можно дополнить нулями) — табл. 1.2.

$$101\ 001_2 = 51_8.$$

Таблица 1.2. **Соответствие двоичных триад восьмеричным цифрам**

Двоичные триады	000	001	010	011	100	101	110	111
Восьмеричные цифры	0	1	2	3	4	5	6	7

Перевод чисел из двоичной системы счисления в шестнадцатеричную (основание — $2^n = 2^4$). Каждый разряд шестнадцатеричного числа содержит 4 бита — тетрада. Разбиваем число на группы по 4 цифры справа налево (дробные — слева направо), а затем преобразуем каждую группу в шестнадцатеричную цифру (последнюю левую можно дополнить нулями) — табл. 1.3.

Таблица 1.3. **Соответствие двоичных тетрад шестнадцатеричным цифрам**

Двоичные триады	0010	1001	1101	0100
Шестнадцатеричные цифры	2	9	D	4

$$101\ 001_2 = 0010\ 1001 = 29_{16}.$$

В табл. 1.4 представлены арифметические действия в двоичной системе счисления.

Таблица 1.4. **Арифметические действия в двоичной системе счисления**

Сложение	Вычитание	Умножение
0 + 0 = 0	0 – 0 = 0	0 · 0 = 0
0 + 1 = 1	0 – 1 = 11	0 · 1 = 0
1 + 0 = 1	1 – 0 = 1	1 · 0 = 0
1 + 1 = 10	1 – 1 = 0	1 · 1 = 1

Примеры:

$$\begin{array}{r} 110_2 \\ +\ 11_2 \\ \hline 1001_2 \end{array} \qquad \begin{array}{r} 110_2 \\ -\ 11_2 \\ \hline 11_2 \end{array} \qquad \begin{array}{r} 110_2 \\ *\ 11_2 \\ \hline 110 \\ 110 \\ \hline 10010_2 \end{array} \qquad \begin{array}{r|l} 110_2 & 11_2 \\ \hline 11_2 & 10_2 \\ \hline 0 & \end{array}$$

1.7. Логические основы ЭВМ

В основе современной логики как науки о формах и способах мышления лежат труды древнегреческих ученых. Аристотель заложил основы формальной логики (отделил формы мышления от его содержания). С помощью логики можно, не вдаваясь в содержание, строить формальные модели окружающего мира.

Основные формы мышления: понятие, суждение, умозаключение. *Понятие* — форма мышления, выделяющая существенные признаки объектов, отличных от других объектов. Объекты, объединенные понятием, образуют множество. Например, понятие «компьютер» — множество электронных устройств. Понятие имеет две стороны: содержание и объем. *Высказывание* (суждение) — форма мышления, в которой что-либо утверждается или

отрицается о реальных предметах, их свойствах и отношениях между ними. Высказывание может быть либо истинно, либо ложно. Истинно — связь понятий правильно отражает свойства и отношения реальных вещей. Высказывание не может быть повелительным или вопросительным. На основе простых можно построить *составные высказывания*. Истинность или ложность простых высказываний устанавливается на основе здравого смысла, а составных — на основе *алгебры высказываний*. *Умозаключение* — форма мышления, с помощью которой из одного или нескольких суждений (посылок) может быть получено новое суждение. Посылками по правилам формальной логики могут быть только истинные суждения.

Булева алгебра (алгебра логики, алгебра высказываний). В алгебре высказываний определяется истинность или ложность составных высказываний без учета их содержания. Высказывания обозначаются именами логических переменных, которые могут принимать лишь два значения: истина (1) либо ложь (0).

Примеры: $A = 2 \cdot 2 = 4 \rightarrow A = 1$; $B = 2 \cdot 2 = 5 \rightarrow B = 0$.

Над высказываниями можно производить *логические операции* для получения новых высказываний. Для этого используют базовые логические элементы, выражаемые словами И, ИЛИ, НЕ.

Логическое умножение (*конъюнкция*) — объединение двух (или нескольких) высказываний в одно с помощью союза И. Обозначения: ^, &, *. $F = A \,\&\, B$, где A и B — логические переменные, F — составное высказывание, которое может принимать значение 1 или 0. Значение логической функции можно определить с помощью *таблицы истинности*, показывающей, какие значения принимает логическая функция при всех возможных наборах ее аргументов (табл. 1.5).

Таблица 1.5. **Таблица истинности для логического умножения**

A	B	$F = A \,\&\, B$
0	0	0
0	1	0
1	0	0
1	1	1

Правило. Составное высказывание, образованное в результате конъюнкции, истинно тогда и только тогда, когда истинны входящие в него простые высказывания.

Логическое сложение (*дизъюнкция*) — объединение двух (или нескольких) высказываний в одно с помощью союза ИЛИ. Обозначения: $\vee$, $+$. $F = A \vee B$. Таблица истинности имеет следующий вид (табл. 1.6).

Таблица 1.6. **Таблица истинности для логического сложения**

A	B	$F = A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Правило. Составное высказывание, образованное в результате дизъюнкции, истинно тогда, когда истинно хотя бы одно из входящих в него простых высказываний.

Логическое отрицание (*инверсия*) — присоединение частицы НЕ к высказыванию. Обозначение: $\overline{A}$. $F = \overline{A}$. Таблица истинности имеет следующий вид (табл. 1.7).

Таблица 1.7. **Таблица истинности для логического отрицания**

A	$F = \overline{A}$
0	1
1	0

Правило. Инверсия делает истинное высказывание ложным и наоборот.

Логическое следование (*импликация*) образуется соединением двух высказываний в одно с помощью конструкции «если..., то...». Когда из истинной предпосылки следует ложный вывод, то составное высказывание ложно.

Работа с таблицей истинности. Каждое составное высказывание можно выразить в виде формулы (логического выражения), в которое входят логические переменные (высказывания) и знаки логических операций, обозначающие логические функции. Примеры: $2 \cdot 2 = 5$, $A = 0$; $2 \cdot 2 \neq 5$, $B = 1$. Порядок выполнения операций: инверсия, конъюнкция, дизъюнкция. Для изменения порядка используют скобки. Для построения таблицы истинно-

сти составного высказывания необходимо определить количество строк, которое равно количеству возможных комбинаций значений логических переменных. Если количество логических переменных n, то количество строк равно 2^n. Например, в формуле две переменные, следовательно, $n = 4$. Количество столбцов равно количеству переменных n плюс количество логических операций. В формуле $F = (A \vee B) \& (\overline{A} \vee B)$ пять операций, следовательно, количество столбцов 2 + 5 = 7 (табл. 1.8).

Таблица 1.8. **Таблица истинности для формулы**

A	B	$A \vee B$	$\overline{A}$	B	$\overline{A} \vee B$	$(A \vee B) \& (\overline{A} \vee B)$
0	0	0	1	1	1	0
0	1	1	1	0	1	1
1	0	1	0	1	1	1
1	1	1	0	0	0	0

Заполним таблицу, выполняя базовые логические операции в соответствии с их таблицей истинности и определим значение F для любого набора переменных.

Логические выражения, у которых таблицы истинности совпадают, называются равносильными (=).

Логические законы: тождества, непротиворечия, исключенного третьего, двойного отрицания, законы Моргана, правило коммутативности, правило ассоциативности, правило дистрибутивности.

Базовые логические элементы ЭВМ. Любые устройства компьютера могут быть собраны из базовых логических элементов И, ИЛИ, НЕ. Эти элементы оперируют с сигналами — электрическими импульсами.

Есть импульс — логическое значение сигнала 1, нет — 0. На входе: сигналы-аргументы. На выходе: сигнал-функция. Преобразование сигнала логическим элементом задается таблицей состояния (таблицей истинности).

Сумматор двоичных чисел — главная часть процессора, обеспечивающая сложение двоичных чисел (рис. 1.2). При сложении двоичных чисел возможен перенос в старший разряд.

Обозначим: A, B — слагаемые; P — перенос; S — сумма.

Ниже представлена таблица сложения одноразрядных двоичных чисел с учетом переноса в старший разряд (табл. 1.9).

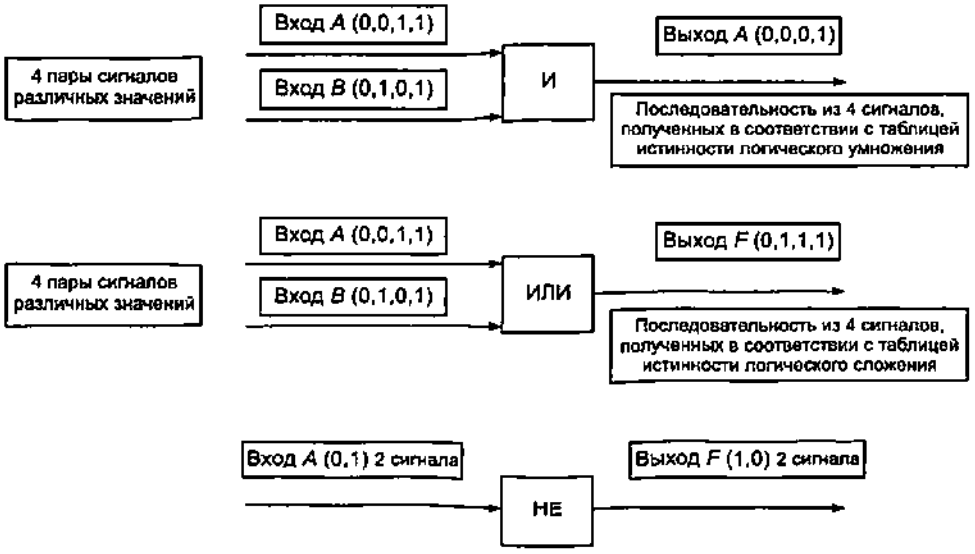


Рис. 1.2. Схема сумматора двоичных чисел

Таблица 1.9. **Сложение одноразрядных двоичных чисел**

Сложение		Перенос	Сумма
A	*B*	*P*	*S*
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$S = (A \vee B)\ \&\ (A\ \&\ B)$ — на основе этой формулы строят схему полусумматора.

Перенос можно реализовать с помощью логического умножения: $P = A\ \&\ B$.

Многоразрядный сумматор процессора состоит из полных одноразрядных сумматоров.

Полный сумматор имеет три входа — A, B, P — и два выхода — S и P.

$$P = (A\ \&\ B) \vee (A\ \&\ P_0) \vee (B\ \&\ P_0);$$

$$S = (A \vee B \vee P_0)\ \&\ P_0 \vee (A\ \&\ B\ \&\ P_0).$$

На рис. 1.3 приводится схема полусумматора, которая реализует суммирование одноразрядных двоичных чисел без учета переноса из младшего разряда.

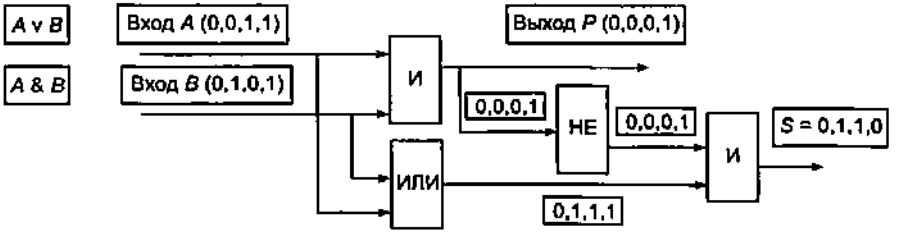


Рис. 1.3. Схема полусумматора двоичных чисел

Триггер — важнейшая структурная единица ОП и внутренних регистров компьютера (рис. 1.4). Это устройство для запоминания, хранения и считывания информации. Каждый триггер хранит 1 бит информации. Триггер можно построить из двух логических элементов: ИЛИ, НЕ.

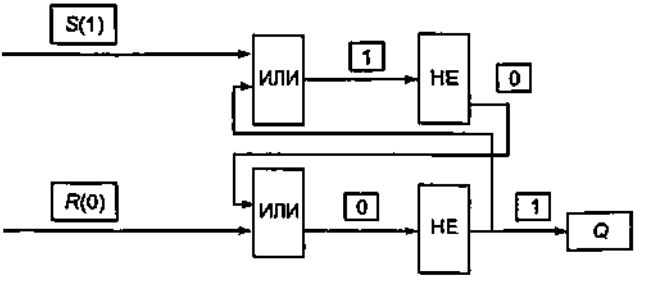


Рис. 1.4. Схема триггера

Дешифратор — устройство для расшифровки (декодирования) сообщения и перевода содержащейся в нем информации на язык воспринимающей системы. Дешифратор имеет *n* входов и *m* выходов. Поступающая на входе информация преобразуется, и на соответствующем выходе (группе выходов) выделяется сигнал, указывающий признак входной информации. В вычислительной технике дешифратор применяют в качестве преобразователя кодов в эквивалентные или непрерывные величины (например, электрический ток, напряжение, угол поворота и др.). Линейный, или одноступенчатый, дешифратор преобразует параллельный двоичный код в унитарный, т. е. позиционный код

(обычно эквивалентный десятичному). Номер вывода соответствует обычно десятичному эквиваленту двоичного кода. Например, количество входов — 2, количество выходов — 8. Ниже приведена таблица состояний дешифратора (табл. 1.10).

Таблица 1.10. **Состояния дешифратора**

X_3	X_2	X_1	Z_0	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	1	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Состоянию каждой функции соответствует только один минтерм, следовательно, функции не нужно минимизировать. Из этого следует, что для реализации полного дешифратора на m входов потребуется $n = 2^m$ элементов конъюнкции и m элементов отрицания. Бывают дешифраторы пирамидальные, двухступенчатые (на интегральных схемах) и др.

Пример выполнения работы «Системы счисления и логические элементы ЭВМ»

1. **Цель работы.** Цель работы — получить практические навыки перевода чисел из одной системы счисления в другую. Следует научиться выполнять арифметические действия в двоичной системе счисления, проектировать из основных логических элементов логические узлы ЭВМ, определять математические выражения и их значения по таблицам истинности.

2. **Содержание работы.**

2.1. Выполнить в соответствии с заданием запись чисел в развернутой форме и перевод чисел:

- из двоичной, восьмеричной и шестнадцатеричной систем счисления в десятичную;

- из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную;
- из двоичной системы счисления в восьмеричную, шестнадцатеричную и обратно.

2.2. Выполнить в соответствии с заданием арифметические операции в позиционных системах счисления.

2.3. Выполнить по индивидуальному заданию построение:

- таблицы истинности заданной логической функции;
- схемы заданного логического элемента;
- схемы заданного логического узла ЭВМ.

2.4. Ответить на контрольные вопросы.

3. Исходные данные.

3.1. Запишите числа 1999_{10} и $19{,}99_{10}$ в десятичной системе в развернутой форме.

3.2. Переведите числа:

- 1011_2, $15{,}5_8$, $1A_{16}$ в десятичную систему счисления;
- 21_{10} в двоичную, восьмеричную и шестнадцатеричную системы счисления;
- 101_2 в восьмеричную и шестнадцатеричную системы счисления.

3.3. Выполните следующие арифметические действия:

$$126_8 + 662_8 = ?;\ 101101_2 + 11011_2 = ?;\ 1C_{16} \cdot A_{16} = ?.$$

3.4. Запишите составное высказывание $2 \cdot 2 = 4$ и $3 \cdot 3 = 10$ в логической форме и по таблице истинности определите значение логической функции.

3.5. Приведите схему базового логического элемента И и схему логического узла ЭВМ, соответствующего формуле $F = (A, B) = -F(A \text{ и } B)$ или Z.

4. Перечень применяемых приборов и материалов:

- ПЭВМ;
- программное обеспечение: ОС Windows XP, программа «Калькулятор».

5. Ход и результаты работы.

5.1. Развернутая форма чисел:

$$1999_{10} = 1 \cdot 10^3 + 9 \cdot 10^2 + 9 \cdot 10^1 + 9 \cdot 10^0;$$

$$19{,}99_{10} = 1 \cdot 10^1 + 9 \cdot 10^0 + 9 \cdot 10^{-1} + 9 \cdot 10^{-2}.$$

5.2. Перевод чисел 1011_2, $15{,}5_8$, $1A_{16}$ в десятичную систему счисления:

$$1011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 2 + 1 = 11_{10};$$

$$15{,}5_8 = 1 \cdot 8^1 + 5 \cdot 8^0 + 5 \cdot 8^{-1} = 8 + 5 + 5 \cdot 1/8 = 13{,}625_{10};$$

$$1A_{16} = 1 \cdot 16^1 + A \cdot 16^0 = 16 + 10 \cdot 1 = 26_{10}.$$

5.3. Перевод числа 21_{10} в двоичную (табл. 1.11), восьмеричную (табл. 1.12) и шестнадцатеричную (табл. 1.13) системы счисления.

Таблица 1.11. **Перевод числа 21_{10} в двоичную систему счисления**

Десятичное число (целое частное)	Делитель (основание системы)	Остаток	Цифры двоичного числа
21	2	1	a_0
10	2	0	a_1
5	2	1	a_2
2	2	0	a_3
1	2	1	a_4 ↑

$$21_{10} = 10101_2.$$

Таблица 1.12. **Перевод числа 21_{10} в восьмеричную систему счисления**

Десятичное число	Делитель (основание системы)	Остаток	Цифры восьмеричного числа
21	8	5	a_0
2	8	2	a_1 ↑

$$21_{10} = 25_8.$$

Таблица 1.13. **Перевод числа 21_{10} в шестнадцатеричную систему счисления**

Десятичное число	Делитель (основание системы)	Остаток	Цифры шестнадцатеричного числа
21	16	5	a_0
1	16	2	a_1 ↑

$$21_{10} = 15_{16}.$$

5.4. Перевод числа 101_2 в восьмеричную и шестнадцатеричную системы счисления.

$$101_2 = 5_8.$$

Каждый разряд шестнадцатеричного числа содержит 4 бита, поэтому в соответствии с правилом дополним число нулем слева, получим двоичную тетраду 0101.

$$0101_2 = 5_{16}.$$

5.5. Арифметические действия:

$126_8 + 461_8 = 607_8$; $101101_2 + 11011_2 = 1000\ 000_2$; $1C_{16} \cdot A_{16} = 78_{16}$;

5.6. Запись составного высказывания $2 \cdot 2 = 4$ и $3 \cdot 3 = 10$ в логической форме и определение по таблице истинности его логического значения.

Простое высказывание $2 \cdot 2 = 4$ истинно, т. е. $A = 1$, а простое высказывания $3 \cdot 3 = 10$ ложно, т. е. $B = 0$. Составное высказывание объединено конъюнкцией (логическое умножение И), поэтому функция будет иметь вид $F = A \,\&\, B$. По приведенной ниже таблице истинности функции логического умножения (табл. 1.14) определяем, что $F = 0$, т. е. составное высказывание $2 \cdot 2 = 4$ и $3 \cdot 3 = 10$ — ложно.

Таблица 1.14. **Таблица истинности функции логического умножения**

A	B	$F = A \,\&\, B$
0	0	0
0	1	0
1	0	0
1	1	1

5.7. Запись схемы базового логического элемента И (рис. 1.5) и схемы логического узла ЭВМ, соответствующего формуле $F = (A \,\&\, B) \vee Z$ (рис. 1.6).

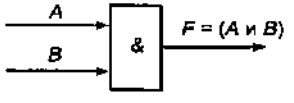


Рис. 1.5. Схема базового логического элемента И

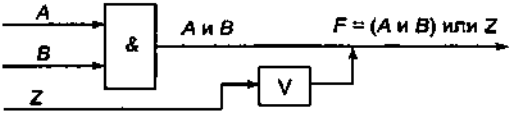


Рис. 1.6. Схема логического узла ЭВМ, соответствующего формуле $F = (A \,\&\, B) \vee Z$

Контрольные вопросы

1. Какое количество цифр используется в K-ичной системе счисления?
2. В чем состоит различие между записями чисел в обычной и экспоненциальной формах и в чем преимущество последней?
3. Что называется основанием системы счисления?
4. Сформулируйте правила перевода чисел из одной системы счисления в другую.
5. Что такое бит, байт, машинное слово?
6. Как определяется истинность или ложность простого и составного высказываний?
7. Чем заменяются высказывания в алгебре высказываний?
8. Каковы базовые логические операции и их обозначения?
9. Как построить таблицу истинности?
10. Каковы функции сумматора, триггера, дешифратора?

Глава 2
ОСНОВЫ РАБОТЫ С ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКОЙ

2.1. Основные этапы развития вычислительной техники

Абак — приспособление, подобное счетам, которое применяли еще древние греки и римляне, а также жители стран Древнего Востока. В XV в. Леонардо да Винчи изобрел суммирующее устройство с зубчатыми колесами для сложения 13-разрядных чисел. В XVII в. немецкие ученые В. Шиккард (1623), Г. Лейбниц (1673) и французский ученый Б. Паскаль (1642) создали механические вычислительные устройства — предшественники всем известного арифмометра. Счетная машина Б. Паскаля могла только складывать и вычитать. Вычислительные машины совершенствовались в течение нескольких веков.

Спустя два столетия, в 1820 г. Шарль Ксавье Томас де Кольмар создал арифмометр — первый многофункциональный калькулятор, который позволял производить умножение, используя принцип Лейбница, и являлся подспорьем пользователю при делении чисел. Это была самая надежная машина в те времена; она не зря занимала место на столах счетоводов Западной Европы. Арифмометр также поставил мировой рекорд по продолжительности продаж: последняя модель была продана в начале XX в.

В 1830 г. английский ученый, профессор математики Кембриджского университета Чарльз Бэббидж обосновал принципы архитектуры вычислительной машины. Работая над проектом аналитической машины — «машины для исчисления разностей», — Ч. Бэббидж предсказал многие идеи и принципы организации и работы современных ЭВМ.

В 1854 г. английский математик Джордж Буль опубликовал книгу «Законы мышления», в которой развил алгебру высказы-

ваний — булеву алгебру. На ее основе в начале 80-х годов XIX в. была построена теория релейно-контактных схем и конструирования сложных дискретных автоматов. Алгебра логики оказала многогранное влияние на развитие вычислительной техники, являясь инструментом разработки и анализа сложных схем, инструментом оптимизации большого числа логических элементов, из многих тысяч которых состоит современная ЭВМ.

Идеи Ч. Бэббиджа реализовал американский ученый Г. Холлерит, который с помощью построенной счетно-аналитической машины и перфокарт за три года обработал результаты переписи населения в США по состоянию на 1890 г. В машине впервые было использовано электричество. В 1896 г. Холлеритом была основана фирма по выпуску вычислительных перфорационных машин и перфокарт.

В 1936 г. английский математик А. Тьюринг ввел понятие «машины Тьюринга» как формального уточнения интуитивного понятия алгоритма. Ученый показал, что любой алгоритм в некотором смысле может быть реализован на машине Тьюринга, а следовательно, доказывал возможность построения универсальной ЭВМ. И та и другая машины аналогично могут быть снабжены исходными данными решаемой задачи и программой ее решения. Машину Тьюринга можно считать как бы идеализированной моделью универсальной ЭВМ.

В 40-х годах XX в. механическая элементная база вычислительных машин стала заменяться электрическими и электронными устройствами. Первые электромеханические машины были созданы в Германии К. Цузе (Ц-3, 1941 г.) и в США под руководством профессора Гарвардского университета Г. Айкена (Марк I, 1944 г.). Первая электронная машина создана в США группой инженеров под руководством доктора Пенсильванского университета Дж. Мочли и аспиранта Дж. Эккерта (ЭНИАК — электронный числовой интегратор и калькулятор, 1946 г.). В 1949 г. в Англии была построена EDSAC — первая машина, обладающая автоматическим программным управлением, внутренним запоминающим устройством и другими необходимыми компонентами современных ЭВМ.

Логические схемы вычислительных машин были разработаны в конце 1940-х годов Дж. фон Нейманом, Г. Гольдстайном и А. В. Берксом. Особый вклад в эту работу внес американский математик Джон фон Нейман, принимавший участие в создании ЭНИАК. Он предложил идею хранения команд управления и

данных в машинной памяти и сформулировал основные принципы построения современных ЭВМ. ЭВМ с хранимой программой оказались более быстродействующими и гибкими, чем ранее созданные. В 1951 г. в США было налажено первое серийное производство электронных машин УНИВАК (универсальная автоматическая вычислительная машина). В это же время фирма IBM начала серийный выпуск машины IBM/701.

В СССР первыми авторами ЭВМ, изобретенной в декабре 1948 г., являются И. С. Брук и Б. И. Рамеев. А первая советская ЭВМ с сохраняющейся программой создана в 1951 г. под руководством С. А. Лебедева (МЭСМ — малая электронная счетная машина). В 1953 г. в Советском Союзе начался серийный выпуск машин, первыми из которых были БЭСМ-1, «Стрела».

С появлением цифровых программно-управляемых машин родилась новая область прикладной математики — программирование, как область науки и профессия она возникла в 1950-х годах.

С 1946 г. и до конца 1950-х годов основным элементом конструкции ЭВМ были электронные лампы (первое поколение). В этот период развитие идеологии и техники программирования шло за счет достижений американских ученых Дж. фон Неймана, сформулировавшего *основные принципы построения ЭВМ*, и Дж. Бэкуса, под руководством которого в 1954 г. был создан Фортран (FORTRAN — FORmula TRANslation) — первый язык программирования высокого уровня, используемый до настоящего времени в разных модификациях.

Дж. фон Нейман в 1945 г. сформулировал основные принципы построения ЭВМ:

1. Принцип хранимой программы.
2. Адресный принцип.
3. Автоматизм.
4. Переадресация (адреса ячеек памяти, указанные в команде, можно вычислять и преобразовывать, как числа).

В соответствии с этими принципами компьютер должен иметь следующие основные конструкции:

- устройство управления, организующее процесс выполнения программ;
- арифметическо-логическое устройство, которое выполняет арифметические и логические операции;
- запоминающее устройство (память) для хранения программ и данных;
- внешние устройства для ввода/вывода информации.

Структурная схема ЭВМ в соответствии с этими принципами представлена на рис. 2.1.

Рис. 2.1. Структурная схема устройства ЭВМ

Как работает компьютер:

1. С какого-либо внешнего устройства в память компьютера вводится программа.

2. Устройство управления считывает содержимое ячейки памяти, где находится первая инструкция (команда) программы, и организует ее выполнение. Далее выполняются другие команды. Устройство управления выполняет инструкции программы автоматически, без вмешательства человека, с помощью ОП и АЛУ.

3. Устройство управления может обмениваться информацией с ОП и УВВ.

4. Результаты выполненной программы должны быть ею выведены на внешние устройства.

Первая в мире ЭВМ ЭНИАК (США) весом 30 т, в которой использовалось 18 тыс. электронных ламп, размещалась на площади 135 кв. м. Достижения в области электроники и микроэлектроники позволили со временем заменять элементную базу ЭВМ на все более совершенную.

В конце 1950-х годов громоздкие электронные лампы заменяют полупроводниками (миниатюрными транзисторами), изобретенными в США в 1948 г. Шокли и Бардиным. Появляются ЭВМ второго поколения. Примерно через 10 лет начинается использование ЭВМ третьего поколения на интегральных схемах, но еще с малой степенью интеграции (микросхема изобретена в 1958 г. Дж. Килби, США). Еще через 10 лет появляются ЭВМ четвертого поколения на больших интегральных схемах (БИС), с большой степенью интеграции, на микропроцессорах. В Японии в 1990-х годах реализованы проекты ЭВМ пятого поколения,

в которых использованы достижения в области искусственного интеллекта и биоэлектроники.

Ниже приведена таблица этапов развития ЭВМ (табл. 2.1).

Если объем оперативного запоминающего устройства (ОЗУ) одной из лучших отечественных машин 1960-х годов М-20, созданной под руководством С. А. Лебедева в 1958 г., составлял 4096 слов (8 Кб) при быстродействии 20 тыс. операций в секунду, то современные персональные компьютеры характеризуются ОЗУ в десятки мегабайт и быстродействием сотни миллионов операций в секунду, что позволяет решать сложнейшие задачи.

Сегодняшние самые быстрые *суперкомпьютеры* могут выполнять сотни миллиардов сложений в секунду. Такое количество арифметических операций под силу выполнить лишь сотням тысяч человек за год. А в разработке находятся компьютеры с быстродействием триллион операций в секунду.

В 1953 г. А. А. Ляпуновым был предложен операторный метод программирования, который заключался в автоматизации программирования, а алгоритм решения задачи представлялся в виде совокупности операторов, образующих логическую схему задачи. Схемы позволяли расчленить громоздкий процесс составления программы, части которой составлялись по формальным правилам, а затем объединялись в целое. Для проверки идей операторного метода в СССР в 1954 г. была разработана первая программирующая программа ПП-1, а в 1955 г. — более совершенная ПП-2. В 1956 г. разработана ПП БЭСМ, в 1957 г. — ППСВ, в 1958 г. — для машины «Стрела».

В США в 1954 г. стал применяться алгебраический подход, совпадающий, по существу, с операторным методом. В 1956 г. корпорацией IBM разработана универсальная ПП «Фортран» для автоматического программирования на ЭВМ IBM/704.

В этот период по мере накопления опыта и теоретического осмысления совершенствовались языки программирования. В 1958—1960 гг. в Европе был создан язык программирования Алгол (Algol), который породил целую серию алголоподобных языков: Алгол-W (1967), Алгол-68, Паскаль (Pascal — Н. Вирт, 1970 г.), Си (С — Д. Ритчи и Б. Керниган, 1972 г.), Ада (Ada — под руководством Ж. Ишбиа, 1979 г.), Си++ (С++ — 1983 г.). В 1961—1962 гг. Дж. Маккарти в Массачусетском технологическом институте создал язык функционального программирования Лисп (Lisp), открывший в программировании одно из альтернативных направлений, предложенных Дж. фон Нейманом.

Таблица 2.1. **Этапы развития ЭВМ**

Этапы	Годы	Элементная база; быстродействие	Отечественные ЭВМ	Зарубежные ЭВМ
I	Середина 40 — середина 50-х годов XX в.	Лампы, ЭЛТ; несколько тысяч операций в секунду	МЭСМ, БЭСМ, «Урал-1»	Серийная ЭВМ IBM-701 (США)
II	Конец 50 — середина 60-х годов XX в.	Транзисторы (намагниченные сердечники); несколько тысяч операций в секунду	БЭСМ-4, М-220, «Урал», «Минск-32»	IBM-7090
III	Конец 60 — начало 70-х годов XX в.	Интегральные полупроводниковые схемы, микропроцессоры	ЕС ЭВМ, мини-ЭВМ, СМ ЭВМ, ЭВМ М-10	IBM-360, IBM-370
IV	2-я половина 70 — начало 80-х годов XX в.	БИС — большие интегральные схемы: 1 кристалл — 1 см^2 содержит сотни тысяч электронных элементов; 10^9 операций в секунду	Многопроцессорные системы, микро-ЭВМ, ПК, супер-ЭВМ (1 млрд оп./с); «Эльбрус» ($1{,}2 \cdot 10^9$ оп./с)	Первый ПК — «Альтаир-8800», Apple Motorola, ПК IBM PC, «Крей-3» (16 млрд оп./с)
IV—V	С конца 80-х годов XX в.	СБИС. Оптические и магнитооптические элементы. Принципиально отличается от предыдущих (от фон-неймановских). Совокупность ЭВМ с высокой производительностью. Общение на естественном языке. Искусственный интеллект. Разговорный язык. Базы знаний	Параллельный суперкомпьютер «Марс» (прототип)	Ведутся разработки, выход планируется в 2011—2012 гг. Перспективными разработками считаются биокомпьютеры, а также квантовые и оптические компьютеры

На начало 1970-х годов существовало более 700 языков высокого уровня и около 300 трансляторов для автоматизации программирования.

Усложнение структуры ЭВМ привело (в 1953 г. для машин второго поколения) к созданию операционных систем (ОС) — специальных управляющих программ для организации и решения задач на ЭВМ

В конце 1980-х годов в Японии и США появились проекты ЭВМ пятого поколения. Прогресс в программировании связан с прогрессом в архитектуре вычислительных систем, отходом от фон-неймановской концепции, с достижениями в области искусственного интеллекта. Революционные изменения в элементной базе ЭВМ связываются с исследованиями по биоэлектронике. В последнее время в Японии удалось создать робота-переводчика, переводящего английскую речь на японский язык и наоборот, осуществляя это голосом человека. Во всех развитых странах работают над комплексами программ для создания роботов для многих сфер человеческой деятельности.

Перспективы развития ЭВМ связывают с разработкой квантовых и оптических компьютеров («замороженный свет»), а также биокомпьютеров на основе ДНК (химическая реакция с участием ферментов, до 330 трлн операций в секунду).

За рубежом в 1960-х годах появились первые вычислительные сети, с которых началась техническая и технологическая революция, так как была предпринята попытка объединить технологию сбора, хранения, передачи и обработки информации на ЭВМ с техникой связи. В Европе в те годы были созданы международные сети EIN и Евронет, затем появились национальные сети. В 1972 г. в Вене была создана сеть МИПСА, к которой присоединились в 1979 г. 17 стран Европы, СССР, США, Канада и Япония. В 1980-х годах в нашей стране была создана система телеобработки статистической информации, обслуживающая государственные и республиканские органы статистики. С 1980-х годов развивается программирование для локальных вычислительных сетей (ЛВС). ЛВС — это коммуникационная система, которая поддерживает в пределах одного здания или некоторой ограниченной территории один или несколько высокоскоростных каналов передачи информации, предоставляемых абонентским системам для кратковременного пользования. К 1990 г. эксплуатировалось свыше 0,5 млн серверов и 5 млн рабочих станций,

работающих под управлением сетевых ОС (например, NetWare компании Novell).

Глобальные вычислительные сети — это сети, использующие информационные ресурсы ЛВС, расположенных на большом расстоянии друг от друга (передача осуществляется с помощью телефонной сети через модемы или по выделенным каналам). Наиболее популярной является сеть Интернет, представляющая собой общемировую совокупность сетей и связывающая миллионы компьютеров.

Сети позволили эффективно использовать аппаратные и программные средства, а также такие многопользовательские системы, как электронная почта, информационные системы на основе баз данных, телеконференции и др. Особой популярностью пользуется система WWW (World Wide Web) — Всемирная паутина, т. е. всемирная распределенная база гипертекстовых документов. Пользователи, используя для программирования язык гипертекстовой разметки HTML, создают свои сайты любой тематики и легко могут получать многообразную информацию, общаться с миллионами пользователей компьютеров. В будущем планируется массовое использование так называемых информационных роботов (Knowbot) — новых систем поиска и обработки информации в Сети, в основе которых имеются уже элементы экспертных систем, позволяющих анализировать искомую информацию и готовить ее для выдачи в форме презентаций. С Интернетом тесно связаны понятия «киберпространство» и «виртуальная реальность». *Киберпространством* называют совокупность всех систем компьютерных коммуникаций и потоков информации, циркулирующих в мировых сетях. *Виртуальная реальность* — фантастический мир, создаваемый на экране компьютера и воспроизводящий образы реального мира и процессы, в нем происходящие. С этими объектами и процессами можно работать как с реальными, проводить различные исследования, имитировать всевозможные ситуации, создавать прекрасные тренажеры для применения полученных навыков в реальности.

2.2. Архитектуры ЭВМ

Для автоматизации работ с данными используют электронные приборы, совокупность которых называют *вычислительной техникой* (*ВТ*). Основным устройством современной ВТ является

компьютер — электронное устройство, предназначенное для автоматизации информационных процессов.

По *принципу назначения* различают большие ЭВМ, мини-ЭВМ, микро-ЭВМ и ПК.

Персональные компьютеры делят на массовые, деловые, портативные, развлекательные и рабочие станции.

Большие ЭВМ (*мэйнфреймы*) — мощные компьютеры, применяемые для решения крупных задач в крупных организациях. Они одновременно могут решать несколько задач для нескольких пользователей по принципу разделения времени. Как правило, такие ЭВМ работают в вычислительных центрах (ВЦ) и обслуживаются большим коллективом специалистов. Структура ВЦ, как правило, включает:

- центральный процессор, обрабатывающий данные и выдающий результаты;
- группу системного программирования, которая разрабатывает, отлаживает и внедряет программное обеспечение для вычислительной системы, обеспечивает программно-аппаратный интерфейс вычислительной системы;
- группу прикладного программирования, которая разрабатывает, отлаживает и внедряет программное обеспечение для обработки конкретных данных для определенных пользователей, обеспечивает пользовательский интерфейс вычислительной системы;
- группу подготовки данных, которая готовит данные для обработки их прикладными программами, включая электронное преобразование (например, сканирование);
- группу технического обеспечения, которая занимается техническим обслуживанием вычислительной системы: ремонтом, подключением новых устройств и т. д.;
- группу информационного обеспечения, которая обеспечивает информацией подразделения ВЦ, создает и хранит архив программ (библиотеки программ) и накопленных данных (банки данных);
- отдел выдачи данных, который получает данные от центрального процессора, преобразует их и выдает в необходимой для заказчика форме.

Мини-ЭВМ меньше по размерам, производительности и стоимости, чем большие. Их применяют для управления производственными процессами, в научной и учебной работе, для

экономических и бухгалтерских расчетов. Такие ЭВМ работают также в небольших ВЦ.

Микро-ЭВМ — компьютеры, обычно работающие в рамках небольшой вычислительной группы, в которую входят также программисты, занимающиеся внедрением приобретенных системных и прикладных программ для нужд организации. В крупных ВЦ микро-ЭВМ используют для выполнения вспомогательных операций.

Персональные компьютеры (ПК) — малые по размерам, но обладающие высокой производительностью машины, предназначенные для обслуживания одного рабочего места, как правило, с одним пользователем. Используют на малых предприятиях и дома. Широко используют в Интернете для получения различного рода информации, обучения, организации досуга и т. д. ПК делят на бытовые и профессиональные.

По *принципу специализации* различают универсальные и специализированные компьютеры. В основе вычислительных систем разного состава (конфигурации) и назначения, как правило, находятся универсальные компьютеры. Специализированные компьютеры решают определенные задачи (например, графические станции), а если специализированные компьютеры одной организации объединяют в сеть, то называют их файловыми серверами. Сетевыми серверами являются компьютеры, которые обеспечивают передачу информации по глобальной сети.

По *размеру* различают настольные, портативные (ноутбуки) и карманные ПК.

По *принципу совместимости* различают аппаратные платформы (например, IBM PC, Apple Macintosh), уровень операционной системы, уровень данных, программную совместимость.

Существует также классификация по *типу процессора* — основного блока или специальной микросхемы (в ПК), который выполняет все вычисления.

Архитектура ЭВМ — комплекс аппаратных и программных средств, с помощью которых обеспечивается программирование и выполнение задач. Архитектура подразделяется на внешнюю и внутреннюю (ЭВМ). В основу положен модульно-магистральный принцип. Модульный принцип позволяет комплектовать нужную конфигурацию и модернизировать ее. Этот принцип опирается на магистральный (шинный) принцип обмена информацией.

Сложность архитектуры компьютера определяет объем набора системных команд. Чем шире набор, тем сложнее архитектура, тем длиннее формальная запись команды (в байтах), тем выше средняя продолжительность исполнения команды, измеренная в тактах работы процессора. Система команд процессоров семейства *Pentium* в настоящее время насчитывает более тысячи различных команд. Такие процессоры называют *процессорами с расширенной системой команд — CISC-процессорами* (*CISC — Complex Instruction Set Computing*).

В противоположность CISC-процессорам в середине 80-х годов появились процессоры архитектуры *RISC* (с сокращенной системой команд (*RISC — Reduced Instruction Set Computing*)). В этой архитектуре количество команд в системе намного меньше и выполняются они намного быстрее. Недостаток в том, что при сокращенном наборе команд сложные операции нужно эмулировать далеко не эффективной последовательностью простейших команд сокращенного набора.

CISC-процессоры используют в универсальных вычислительных системах. RISC-процессоры используют в специализированных вычислительных системах или устройствах, ориентированных на выполнение единообразных операций. ПК *IBM PC* ориентированы на использование CISC-процессоров.

2.3. Принципы работы вычислительной системы

Состав (*конфигурация*) вычислительной системы включает аппаратные и программные средства. *Аппаратное обеспечение* включает устройства и приборы. Современные компьютеры и вычислительные комплексы имеют блочно-модульную конструкцию, что позволяет для разных работ собирать разную конфигурацию из готовых узлов и блоков. Относительно расположения от *центрального процессорного устройства* (*ЦПУ— Central Processing Unit, CPU*) различают *внутренние* и *внешние* устройства. Внешними, как правило, являются большинство устройств ввода-вывода данных (*периферийные* устройства) и внешние носители для хранения данных.

Согласование между отдельными узлами и блоками выполняют с помощью переходных аппаратно-логических устройств, называемых *аппаратными интерфейсами*, стандарты на которые называют *протоколами* — совокупностью технических условий,

необходимых для успешного согласования работы с другими устройствами.

Интерфейсы в архитектуре вычислительной системы условно делят на две большие группы: *последовательные* и *параллельные*, которые обеспечивают последовательную, бит за битом, передачу данных или параллельную — одновременно группами битов. Производительность параллельных интерфейсов измеряют *байтами в секунду* (байт/с; Кб/с; Мб/с). Обмен данными через последовательные устройства производится не байтами, а битами, их производительность измеряют битами в секунду (бит/с; Кбит/с; Мбит/с). Иногда скорость последовательных устройств выражают в *знаках в секунду*, или, что то же самое, в *символах в секунду* (с/с), но эта величина имеет не технический, а справочный, потребительский характер. Современные высокоскоростные последовательные интерфейсы не уступают параллельным, а нередко и превосходят их по пропускной способности.

Компьютеры работают, используя наборы инструкций, называемые компьютерными программами. Программы, выполняющиеся на компьютере, называются программным обеспечением.

Программное обеспечение предназначено для управления аппаратными средствами и обработки данных. Программное и аппаратное обеспечение вычислительной системы работают в неразрывной связи и непрерывном взаимодействии. Между программами, как и между физическими узлами и блоками, существует взаимосвязь — многие программы работают, опираясь на другие программы более низкого уровня, т. е. можно говорить о *межпрограммном интерфейсе*. Такой интерфейс основан на существовании технических условий и протоколов взаимодействия, а на практике он обеспечивается распределением программного обеспечения на несколько взаимодействующих уровней. Каждый следующий уровень опирается на программное обеспечение предыдущих уровней: прикладное — служебное — системное — базовое.

Самый низкий уровень программного обеспечения — *базовое ПО*. Оно отвечает за взаимодействие с *базовыми аппаратными средствами*. Базовые программные средства непосредственно входят в состав базового оборудования и хранятся в специальных микросхемах, называемых *постоянными запоминающими устройствами* (*ПЗУ — Read Only Memory, ROM*). Программы и данные записываются («прошиваются») в микросхемы ПЗУ на этапе производства и при эксплуатации не могут быть изменены. Когда

изменение базовых программных средств во время эксплуатации является целесообразным, вместо микросхем ПЗУ применяют *перепрограммируемые постоянные запоминающие устройства* (*ППЗУ — Erasable and Programmable Read Only Memory, EPROM*).

Системное ПО — переходный уровень. Программы системного уровня обеспечивают взаимодействие прочих программ с программами базового уровня и с аппаратным обеспечением, как бы управляют оборудованием и решением задач на нем. Программы, отвечающие за взаимодействие с конкретными устройствами, называются *драйверами устройств* — они входят в состав ПО системного уровня.

Другой класс программ системного уровня отвечает за взаимодействие с пользователем. Эти программы называют *средствами обеспечения пользовательского интерфейса*. От них зависит удобство работы с компьютером и производительность труда на рабочем месте.

Совокупность ПО системного уровня образует *ядро операционной системы компьютера*. Понятие операционной системы (ОС) рассмотрим позже. Здесь отметим, что наличие ядра операционной системы — непременное условие для возможности работы человека с вычислительной системой.

Служебное ПО — уровень взаимодействия с программами как базового, так и системного уровня. Назначение служебных программ (*утилит*) состоит в автоматизации работ по проверке, наладке и настройке компьютерной системы. Во многих случаях они используются для расширения или улучшения функций системных программ. Некоторые служебные программы (обслуживания) изначально включают в состав ОС, но большинство служебных программ являются внешними и служат для расширения функций ОС.

Прикладное ПО — уровень решения конкретных задач в различных сферах деятельности. Назовем самые распространенные программные комплексы: текстовые редакторы и процессоры, графические редакторы, СУБД, электронные таблицы, САПР (CAD-системы), настольные издательские системы, экспертные системы, веб-редакторы, браузеры (обозреватели, средства просмотра веб-страниц), интегрированные системы делопроизводства, бухгалтерские и финансовые аналитические системы, геоинформационные системы (ГИС), системы видеомонтажа и т. д.

Между прикладным программным обеспечением и системным существует непосредственная взаимосвязь (первое опирает-

ся на второе), поэтому можно утверждать, что универсальность вычислительной системы, доступность прикладного ПО и возможности компьютера зависят от типа ОС, от ее способности обеспечить взаимодействие человека, программ и оборудования.

2.4. Состав и назначение основных элементов персонального компьютера

В состав базовой конфигурации ПК входят в настоящее время, как правило, четыре устройства: системный блок, дисплей, клавиатура, мышь. К базовой конфигурации можно подключать различные периферийные устройства: принтеры, накопители данных, различные манипуляторы (джойстик, трекбол, световое перо), устройства оптического считывания изображений (сканеры), графопостроители и др. Эти устройства подсоединяются к системному блоку с помощью кабелей через гнезда (разъемы), размещенные обычно на задней стенке системного блока. Иногда дополнительные устройства вставляются в системный блок (например, модем, стример). Все модули ПК связаны с системной магистралью (шинами).

Системный блок — главный блок, включающий центральный микропроцессор, сопроцессор, модули оперативной и постоянной памяти, контроллеры, накопители на МД и другие функциональные модули. По форме корпуса, в котором расположены все узлы, ПК подразделяют на горизонтальные (Desktop) и вертикальные (Tower). Кроме формы, требования к размещаемым устройствам определяет так называемый форм-фактор — определяющий стандарт корпуса ПК. Известны форм-факторы AT, ATX и др. Форм-фактор корпуса должен быть согласован с форм-фактором системной (материнской) платы, которая является главной платой ПК. Также немаловажен подбор подходящего блока питания.

На материнской плате расположены: процессор (основная микросхема), микропроцессорный комплект (набор микросхем), магистраль (шины), оперативная память, постоянное запоминающее устройство (ПЗУ), разъемы (порты) для подключения устройств.

Центральный микропроцессор (*ЦМП*) — обрабатывает информацию разного вида, поступающую в виде электрических импульсов, которые записываются в виде последовательности 0 и 1.

ЦМП включает арифметически-логическое устройство (АЛУ), память (регистры), устройство управления (УУ) и схемы управления шиной, кэш-память. В системный блок может быть встроено звуковое устройство, с помощью которого удобно следить за работой машины. Со звуковым устройством часто связан таймер, позволяющий вести отсчет времени работы ПЭВМ, решения задач, календарного времени. Микропроцессор — ядро ПЭВМ, он выполняет функции обработки информации и управления всех блоков. Как правило, он выполнен на одном кристалле (на одной СБИС).

В АЛУ арифметические операции выполняются в двоичной системе счисления и в двоично-десятичном коде. Для выполнения арифметических операций над числами с плавающей точкой имеется специальный блок — арифметический *сопроцессор* (имеет собственные регистры данных и управления и работает параллельно с микропроцессором). В табл. 2.2 представлены основные характеристики микропроцессоров фирмы Intel. На рис. 2.2 изображена схема устройства компьютера.

Рис. 2.2. Схема устройства компьютера

Таблица 2.2. **Основные характеристики некоторых моделей МП фирмы Intel**

Характеристика	Модель			
	Intel Sx2	Intel Dx2	Intel Dx4	Pentium
Ширина шины (бит)	32	32	32	32
Инструкция на такт (max)	1	1	1	2
Физическое и адресное пространство (Гб)	4	4	4	4
Рабочая частота (МГц)	50	40, 50, 66	75, 100	60, 66, 90—200
Математический сопроцессор	Процессор Over Drive	Встроен	Встроен	Встроен
Кэш-память (Кб)	8	8	16	8 (данные), 8 (код)

Устройство управления (*УУ*) микропроцессорного типа обеспечивает обработку данных с помощью блока очереди команд. Этот блок заполняет очередь команд длиной 32 бит. Выборка байтов из памяти выполняется в промежутках между магистральными циклами команд.

Внутренняя память состоит из оперативной (ОП) и постоянной (ПП). Она построена на БИС или СБИС. В ОП имеется стек (выделяемая область). Используется при организации прерываний и обращений к подпрограммам. От ОП зависит, с какими программами можно работать. Для ускорения доступа к ОП используется кэш-память.

Оперативная память (*RAM — Random Access Memory*) — это массив кристаллических ячеек, способных хранить данные. ОП в компьютере размещается на стандартных панельках — модулях. Модули ОП вставляют в соответствующие разъемы на материнской плате.

С точки зрения физического принципа действия различают динамическую память (DRAM) и статическую память (SRAM).

Ячейки динамической памяти (*DRAM*) можно представить в виде микроконденсаторов, способных накапливать заряд на своих обкладках.

Ячейки статической памяти (*SRAM*) можно представить как электронные микроэлементы — триггеры, состоящие из нескольких транзисторов. В триггере хранится не заряд, а состояние (включен/выключен), поэтому этот тип памяти обеспечивает бы-

стродействие. Микросхемы динамической памяти используют в качестве основной оперативной. Микросхемы статической памяти используют в качестве вспомогательной памяти (так называемой кэш-памяти), предназначенной для оптимизации работы процессора.

Адрес каждой ячейки памяти выражается числом, в большинстве процессоров предельный размер адреса обычно составляет 32 разряда, поэтому независимых адресов может быть 2^{32}. Одна адресуемая ячейка содержит восемь двоичных ячеек, в которых можно сохранить 8 бит (байт) данных. В современных компьютерах возможна *непосредственная адресация* размером 2^{32} байт = 4 Гб. Но предельный размер поля оперативной памяти в компьютере определяется микропроцессорным комплектом (*чипсетом*) материнской платы и обычно не может быть больше нескольких гигабайт. Минимальный объем памяти составляет 128 Мб. Сегодня типичным считается размер оперативной памяти в 256—512 Мб и более. Скорость передачи данных определяет максимальную пропускную способность памяти (в мегабайтах или гигабайтах в секунду) в оптимальном режиме доступа. Одинаковые по объему модули могут иметь разные скоростные характеристики. *Время доступа* измеряется в миллиардных долях секунды (*наносекундах, нс*). Для современных модулей памяти это значение составляет 5+ нс.

ПП используется для хранения системных программ, в частности BIOS (Basic Input and Output System) — базовой системы ввода/вывода, вспомогательных программ и т. д. Они предназначены для постоянного исполнения. Процессору нужны команды уже в момент включения. Поэтому сразу после включения на адресной шине аппаратно выставляется стартовый адрес, а далее ПК начинает работать по программам. Стартовый адрес указывает не на пустую пока ОП, а на ПЗУ — постоянное запоминающее устройство. Микросхема ПЗУ может хранить информацию и при выключенном ПК. «Зашитые» в ПЗУ программы записывают при изготовлении микросхемы, они являются базовой системой ввода/вывода — BIOS, которая проверяет состав и работоспособность ПК и обеспечение взаимодействия с клавиатурой, монитором, накопителями на жестких (НЖМД) и гибких (НГМД) магнитных дисках.

На материнской плате есть также энергонезависимая память — CMOS, — содержимое которой не стирается при выключении ПК. Данные в CMOS можно заносить и изменять самостоятельно в соответствии с тем, какое оборудование входит в

состав системы. Эта микросхема подпитывается от элемента питания на материнской плате.

Кэш-память — внутренняя память, используемая для буферизации часто выполняемых команд. Ее объем примерно 8 Кб, она повышает производительность МП.

Сегментация памяти — средство управления пространством логических адресов. Сегментированная память имеет признаки расположения, размера, типа. В состав внутренней памяти МП входят функциональные регистры[1]: общего назначения, указатель команд, флагов и сегментов. Восемь 32-разрядных регистров общего назначения используются для хранения данных и адресов. Они обеспечивают работу с данными разрядностью 1, 8, 16, 32 и 64 бит с адресами 16 и 32 бит. Каждый регистр имеет имя (ESP и др.), 32-разрядный указатель команд (указатель адреса), 32-разрядный регистр флагов, который указывает признаки результата выполнения команды. Разряды сегментов — адресуемые сегменты памяти. УУ МП обеспечивает *многозадачность.*

Защита памяти от несанкционированного доступа осуществляется с помощью привилегий. Уровни привилегий — от 0 до 3. Наиболее защищенная область памяти отводится под ядро ОС, имеет уровень 0.

Порты бывают последовательные и параллельные. Через первый информация передается поразрядно, по малому числу проводов. По второму — по числу проводов, равному числу разрядов. Последовательная передача более медленная, чем параллельная.

Шинные интерфейсы материнской платы. Шина — коммуникационная линия, предназначенная для передачи данных между компонентами ОС.

ISA (Industry Standard Architecture) — позволяет связать все устройства системного блока между собой и обеспечивает подключение новых устройств через стандартные разъемы (слоты). Пропускная способность — до 5,5 Мб/с.

EISA (Extended ISA) — расширенный ISA, имеет увеличенный разъем и увеличенную производительность (до 32 Мб/с).

VLB (VESA Local Bus) — локальная шина стандарта ESA. Имеет повышенную частоту до 50 МГц, интерфейс для подключения видеоадаптера, пропускную способность до 130 Мб/с.

---

[1] Регистр — узел для приема, промежуточного хранения и выдачи n-разрядных слов. Основным элементом схемы регистра является триггер.

PCI (Peripheral Component Interconnect) — стандарт подключения внешних компонентов. Для связи с основной шиной ПК (ISA/EISA) используются специальные интерфейсные преобразователи — мосты (PCI Bridge). Поддерживает частоту шины 33—66 МГц и пропускную способность 132—264 Мб/с, а также 528 Мб/с (для 64-разрядных процессоров).

FSB (Front Side Bus) — используется, начиная с процессора Intel Pentium Pro, применяется для связи процессора и памяти. Работает на частоте 100—200 МГц.

AGP (Advanced Graphic Port) — усовершенствованный графический порт для видеокарты. Работает на частоте 33—66 МГц, имеет высокую производительность, так как передает несколько сигналов за один такт (например, в режиме AGP4x скорость передачи составляет 1066 Мб/с).

PCMCIA (Personal Computer Memory Card International Association) — стандарт Международной ассоциации производителей плат памяти для портативных ПК.

USB (Universal Serial Bus) — универсальная последовательная магистраль. Стандарт, определяющий способ взаимодействия ПК с периферийным оборудованием. Позволяет подключать до 256 различных устройств, имеющих последовательный интерфейс. Исключает конфликты между различным оборудованием.

Обмен информацией между блоками идет через *магистраль*, включающую три шины: адреса (16, 20, 24, 32 разряда), данных (8, 16, 32 разряда), управления (командная).

Адресная шина. У процессоров семейства Pentium адресная шина 32-разрядная, т. е. состоит из 32 параллельных проводников. Наличие или отсутствие напряжения на какой-то из линий означает, что на этой линии выставлена соответственно единица или нуль. Комбинация из 32 нулей и единиц образует 32-разрядный адрес ячейки ОП. К ней и подключается процессор для копирования данных из ячейки в один из своих регистров.

Шина данных. По этой шине происходит копирование данных из ОП в регистры процессора и обратно. В современных ПК шина данных, как правило, 64-разрядная, т. е. состоит из 64 линий, по которым за один раз на обработку поступают сразу 8 байт.

Шина управления. Процессор должен знать, что следует сделать с байтами, хранящимися в его регистрах. Команды управления поступают в процессор из тех областей ОП, где хранятся не массивы данных, а программы.

Совокупность всех команд, которые может выполнить процессор над данными, образует так называемую *систему команд процессора*. Процессоры, относящиеся к одному семейству, имеют одинаковые или близкие системы команд. Если они имеют одинаковую систему команд, то они полностью совместимы на программном уровне. Это означает, что программа, написанная для одного процессора, может исполняться и другим. Процессоры, имеющие разные системы команд, как правило, несовместимы или ограниченно совместимы на программном уровне.

Накопители — предназначены для долговременного хранения информации.

Жесткий диск (*НЖМД*) — устройство для долговременного хранения информации, представляющее собой группу соосных магнитных дисков, вращающихся с большой скоростью (до 250 об./с). На каждой из $2n$ поверхностей расположена головка для чтения/записи данных. Управляет работой диска контроллер (микросхема) жесткого диска. Основными характеристиками ЖМД являются емкость и производительность (скорость внутренней передачи данных). К настоящему времени емкость достигает 40 Гб и более (достигает 3000 Гб — 3 Терабайт), а производительность 30—100 Мб/с. Учитывают также параметр среднего времени доступа, который определяет время поиска необходимых данных и зависит от скорости вращения диска (от 10 до 4 мкс при 5400—10000 об./мин и более).

Дисковод гибких дисков (*НГМД*) — специальный накопитель, предназначенный для гибких магнитных дисков (дискет), на которых можно хранить и оперативно переносить небольшие объемы информации. Приемное отверстие накопителя для дискет находится на лицевой стороне корпуса системного блока.

Дисковод компакт-дисков CD-ROM. CD-ROM (Compact Disc Read-Only Memory) — постоянное запоминающее устройство на основе компакт-диска. Принцип действия этого устройства состоит в считывании числовых данных с помощью лазерного луча, отражающегося от поверхности диска. Дисководы CD-ROM относят к аппаратным средствам мультимедиа.

Стример — ленточный магнитный накопитель, запоминающее устройство. Основное назначение: запись и воспроизведение информации, архивация и копирование данных. Емкость современных стримеров — 60—640 Гб.

Внешние запоминающие устройства рассмотрим в разделе 2.6.

Контроллеры служат для управления внешними устройствами (ВУ). Каждому ВУ соответствует свой контроллер — печатная плата, вставляемая внутрь системного блока. Такие платы часто называют *адаптерами* ВУ. Адаптер имеет регистр состояния и регистр данных. Работает по команде от МП автономно. Эти регистры — порты ввода/вывода (имеют адреса).

Видеоадаптер (*видеокарта*). Вместе с дисплеем *видеокарта* образует *видеоподсистему ПЭВМ*. Ранее в ПК существовала выделенная *экранная область памяти*, в которую процессор помещал данные об изображении. Специальный *контроллер экрана* считывал данные о яркости точек экрана из ячеек памяти этой области и в соответствии с ними управлял разверткой горизонтального луча электронной пушки монитора. Постепенно процессор перестал справляться с построением и обновлением изображения. Тогда и появился блок, управляющий экраном, — *видеоадаптер*. Это отдельная дочерняя плата, которая вставляется в один из слотов материнской платы и называется видеокартой, выполняющей функции *видеоконтроллера*, *видеопроцессора* и *видеопамяти*. Виды видеоадаптеров: MDA (монохромный), CGA (4 цвета), EGA (16 цветов), VGA (256 цветов), SVGA (16,7 млн цветов). Для последнего имеется выбор разрешения экрана: 640 × 480, 800 × 600, 1024 × 768, 1152 × 864, 1280 × 1024 точек (пикселей) и т. д. Так, например, для экранов WHUXGA разрешение достигает 7680 × 4800. Это позволяет при выборе размера экрана учитывать оптимальное разрешение экрана. Так, минимальное разрешение экрана для современных программ должно быть 800 × 600 точек, что обеспечивает размер экрана (ЭЛТ или ЖК) не менее 15 дюймов; типичное разрешение для современных программ обеспечивает размер экрана 17—19 дюймов (1024 × 768 и 1280 × 1024 точек), 24 дюйма (1920 × 1200). Цветовое разрешение (глубина цвета) определяет количество оттенков точки экрана. Оно зависит от разрешения и видеопамяти. Объем видеопамяти, установленной на видеоадаптер, обычно равен 128—512 Мб и более. Одним из свойств видеоадаптера является видеоускорение — преобразование данных в микросхемах видеоускорителя. Существует два типа видеоускорителей: плоской (2D) и трехмерной (3D) графики. Современные видеокарты имеют функции ускорения обоих типов.

Звуковая карта. Звуковая карта устанавливается в один из разъемов материнской платы в виде дочерней карты и выполняет

вычислительные операции, связанные с обработкой звука, речи, музыки. Звук воспроизводится через внешние звуковые колонки, подключаемые к выходу звуковой карты. Специальный разъем позволяет отправить звуковой сигнал на внешний усилитель. Имеется также разъем для подключения микрофона, что позволяет записывать речь или музыку и сохранять их на ЖМД. Основной параметр звуковой карты — разрядность (количество битов — 16, 32, 64, — используемых при преобразовании сигналов из аналоговой в цифровую форму и наоборот). Чем выше разрядность, тем лучше качество звучания. Стандарты записи-воспроизведения звука:

- Dolby Digitals;
- Dolby Surround;
- обычный стереозвук;
- 4-канальное стерео.

Для получения звука среднего качества используют *интегрированные звуковые системы*, в которых функции обработки выполняются центральным процессором и микросхемами материнской платы. Устройство воспроизведения звука подключается к гнездам, установленным непосредственно на материнской плате.

2.5. Устройства ввода/вывода данных

Внешние устройства ПЭВМ по назначению делятся на устройства ввода, вывода, хранения данных, обмена данными. Их выбор зависит от задач пользователя. ВУ составляют 80 % стоимости ПЭВМ. Базовый набор ВУ: клавиатура, дисплей, НЖМД, НГМД составляют вместе с системным блоком базовую конфигурацию ПЭВМ.

В качестве *устройств ввода* используют клавиатуру, различные манипуляторы (мышь, трекбол и др.), сканеры (R = 8000 точек/дюйм = 300 точек/мм и др.), цифровые камеры, микрофон и др.

Клавиатура реализует диалоговое общение пользователя с ПЭВМ: ввод команд; запись, корректировку и отладку программ; ввод данных и команд в процессе решения задач. Типовые размеры клавиатуры, как правило, 40 × 450 × 180 мм. Состоит она примерно из 100 клавиш, которые можно разделить на 5 групп: буквенно-символьные, символьно-цифровые, управ-

ляющие (**Enter**, **Ctrl**, **Alt**, **Shift**, **Tab**, **Esc**, **Pause**, **Print Screen**, **Caps Lock**, **Num Lock**, **Scroll Lock** — последние три имеют световые индикаторы), функциональные (**F1—F12**), управления курсором (стрелки, **Page Up**, **Page Down**, **Home**, **End**, **Delete**, **Insert**).

Клавиатура ПЭВМ передает МП не код символа, а порядковый номер нажатой клавиши и время нажатия. Интерпретация нажатия выполняется программным путем, поэтому кодировка клавиш не зависит от кодировки символов.

Мышь — механический или оптический манипулятор для управления программами. Бывает 2—3-кнопочной. Функции кнопок зависят от программы. Драйвер мыши интерпретирует сигналы, поступающие через стандартный специальный порт. Принцип управления мыши называют событийным. С помощью мыши пользователь на экране изменяет свойства объектов и заставляет работать элементы управления компьютера.

Для ввода графической информации используют *сканеры*, *графические планшеты* (*дигитайзеры*) и *цифровые фотокамеры*. Сканеры бывают ручные, листовые, планшетные, барабанные, форм, штрих-сканеры.

В качестве *устройств вывода* используют дисплей, принтеры и др.

Дисплей — необходимое устройство визуального вывода и воспроизведения информации на экране. Изображение на экране ЭЛТ получается в результате облучения люминофорного покрытия направленным пучком электронов, разогнанных в вакуумной колбе. Дисплеи бывают алфавитно-цифровые, графические, цветные или монохромные. Для получения цветного изображения люминофорное покрытие имеет светящиеся красные, зеленые и синие точки или полоски. Размер пикселя зависит от размера зерна люминофора, покрывающего экран. Размер зерна может быть 0,39, 0,31, 0,28 мм и меньше.

Основные характеристики:

- текстовый формат — число символов в строке и строк на экране;
- разрешающая способность (R) изображения — число точек по горизонтали и число точечных строк по вертикали;
- параметр яркости — количество уровней яркости в монохромном режиме и количество цветов в цветовом режиме.

Дисплей работает под управлением видеоадаптера — платы с микросхемой, которая находится в системном блоке ПК. Видеоадаптер предусматривает текстовый и графический режимы ра-

боты. Используют в основном цветные дисплеи с видеоадаптерами VGA, AVGA (супер-VGA). Число цветов зависит от типа адаптера, его разрешения и объема видеопамяти.

Размер экрана определяет различимость изображения в целом и четкость его отдельных элементов. Растровая память (Bit Mapping) — 1, 2, 4 бита, каждый элемент изображения формируется из фрагментов растровой памяти. Типы дисплеев: VGA или SVGA (имеют разрешение 1024 × 1024 и 2048 × 2048 точек, 4096 базовых цветов, что равно 16 млн оттенков), ранее использовались CGA, EGA и др.

Класс защиты дисплея должен соответствовать международным стандартам: MPR II, TCO-92, TCO-95, TCO-99. Эти стандарты устанавливают безопасные уровни электромагнитного излучения, эргономические и экологические нормы.

Печатающие устройства: принтеры — матричные, лепестковые, струйные, лазерные, светодиодные, графопостроители; плоттеры — рулонные, планшетные.

Принтеры используют для распечатки текстовой и графической информации. В матричных принтерах печать осуществляется ударом матрицы из иголок через красящую ленту (ресурс годности — 500 листов). Разрешение таких принтеров — примерно 250 точек/дюйм. Струйные и лазерные принтеры могут выполнять цветную печать. Струйный принтер печатает с помощью распыления специальных мелкодисперсных чернил. Струйные принтеры очень чувствительны к применяемым расходным материалам (бумаге и картриджам). Лазерные принтеры чувствительны к правильной закладке бумаги в подающие лотки и использованию оригинальных картриджей.

Передача информации в сеть осуществляется с помощью *модема* и *сетевой платы* по каналам связи.

Модем (МОдулятор + ДЕМодулятор) — устройство, преобразующее в соответствии с протоколом цифровые данные, поступающие из компьютера, путем модуляции (по амплитуде, частоте, фазе) в аналоговую форму и путем демодуляции обратно. Данные передаются по каналам связи (например, по телефонной линии). Каналы связи — это физические линии (проводные, оптоволоконные, кабельные, радиочастотные), которые различаются по способу использования (коммутируемые и выделенные) и способу передачи данных (цифровые или аналоговые сигналы). В зависимости от типа канала связи устройства приема-передачи подразделяют на радиомодемы, кабельные модемы и пр.

Широко применяются модемы, ориентированные на подключение к коммутируемым телефонным каналам.

Акустические системы (звуковые колонки, наушники) — устройства вывода звуковой информации. Для воспроизведения качественного звука левая и правая колонки должны звучать одинаково.

2.6. Запоминающие устройства

Необходимость во *внешних запоминающих устройствах* возникает, когда в вычислительной системе обрабатывается больше данных, чем можно разместить на базовом жестком диске, и когда данные имеют повышенную ценность и необходимо выполнять резервное копирование на внешнее устройство.

Часто требуется перенос информации с одного компьютера на другой. Переносить информацию с одного рабочего места на другое путем переноса жесткого диска теоретически возможно, но это нетехнологично, поскольку требует соблюдения особых условий и квалификации. Поэтому для оперативного переноса небольших объемов информации применяют *гибкие магнитные диски* (*дискеты*), которые вставляют в специальный накопитель (дисковод). Гибкие диски — это пластиковый футляр, содержащий магниточувствительную пленку, на которую записывается информация.

Приемное отверстие накопителя находится на лицевой стороне системного блока. Основными параметрами гибких дисков являются технологический размер (в дюймах), плотность записи (в кратных единицах) и полная емкость. Для первых ПЭВМ фирмы IBM (начало 1980-х годов) использовались односторонние гибкие диски диаметром 5,25 дюйма. емкостью 160 Кб, далее появились аналогичные двусторонние диски емкостью 320 Кб. В 1984 г. появились гибкие диски 5,25 дюйма высокой плотности (1,2 Мб). Сейчас эти диски не используют, не выпускают и дисководы для них. С 1980 г. выпускают диски размером 3,5 дюйма: односторонние емкостью 180 Кб, двусторонние — 360 Кб, двусторонние двойной плотности — 720 Кб, высокой плотности (HD) — 1,44 Мб. Чтобы защитить информацию на ГМД от стирания и перезаписи, нужно сдвинуть задвижку так, чтобы образовалось открытое отверстие. Гибкие диски считаются малонадежными носителями информации. Пыль, грязь, вла-

га, температурные перепады и внешние электромагнитные поля очень часто являются причиной утраты записанных на дискете данных. ГМД используют в основном для транспортировки информации или как дополнительное резервное средство хранения информации. Данные не удаляются с жесткого диска до тех пор, пока потребитель не подтвердил их получение. Как правило, делают две резервные копии на ГМД. До начала работы с данными диск следует проверить антивирусными программами. Среди вредоносных программ есть такие, которые поражают не только файлы программ и данных, но и носители информации. Даже чистый гибкий диск может содержать так называемые загрузочные вирусы. С данными, поставленными на гибком диске, работать небезопасно (для данных) и непроизводительно. Нужно скопировать информацию на жесткий диск компьютера и работать только с жестким диском. В новейших компьютерах происходит постепенный отказ и от этого типа носителей.

Внутри системного блока отдельно закреплены магнитные накопители:

- дисковый накопитель (жесткий диск) — постоянное хранилище программ и данных. Емкость современных жестких дисков достигает 10 Гб и более;
- дисководы для гибких магнитных дисков (ГМД);
- дисководы для CD и DVD.

Кроме того, записать/перезаписать информацию можно на флэш-диски — миниатюризированные диски размером несколько сантиметров, подключаемые к ПК через разъем к порту USB.

Для записи информации на компакт-диски и считывания ее существуют устройства — CD-рекордеры.

Сменные носители (дискеты, компакт-диски, DVD и др.) служат для переноса информации.

Объем памяти флэш-дисков может быть 256 Мб и выше.

Объем памяти компакт-диска диаметром 12 см и толщиной 1,2 мм может быть 640—700 Мб и выше. Диски могут хранить информацию разного вида. Имеются компакт-диски (CD-R — CD Recordable) только для записи и чтения однажды записанной информации. Их называют «записываемые компакт-диски». Так называемые «перезаписываемые диски» (CD-RW — CD Rewritable) используются для многократной записи информации. Информация на них записывается медленнее, чем на CD-R, но объем памяти у CD-RW не меньше. Запись и воспроизведение выполняется с помощью низкочастотного так называемого красного лазера.

Объем памяти DVD достигает 4,7 Гб, что в несколько раз выше, чем у CD. На них в основном записывают цифровую видео- и звуковую информацию. Сейчас уже начали выпускать двухслойные HD-DVD емкостью 30 Гб и планируют к выпуску трехслойные диски емкостью 45 Гб.

В перспективе — использование носителей нового поколения, использующих технологию красного лазера, которые могут работать с имеющейся техникой: FVD (Forward Versatile Disc) емкостью 6 Гб и двухслойные FVD — 11 Гб.

Планируется использовать синие диски BD (Blue-ray Disk) размером 12 см и объемом памяти 25 Гб, производимые на основе высокочастотных сине-фиолетовых лазеров. Корпорация Sony разработала для формата Blue-ray Disk дисковый носитель информации с основой из легкого картона, который должен заменить обычно используемый поликарбонатный пластик. Переход на более коротковолновый лазер (405 нм) позволяет плотнее размещать информацию на диске, умещая на один слой до 27 Гб данных. Имеется спецификация двухслойных BD-дисков емкостью 46,6 и 50 Гб. Компания Toshiba выпустила четырехслойный диск емкостью 100 Гб.

Стримеры — накопители на магнитной ленте, могут быть внешними. Их отличает низкая цена, но малая производительность и недостаточная надежность. Испытывают электромагнитные и повышенные механические нагрузки и могут часто выходить из строя. Емкость магнитных кассет (картриджей) для стримеров достигает нескольких десятков гигабайт. Дальнейшее повышение емкости за счет повышения плотности записи снижает надежность хранения, а повышение емкости за счет увеличения длины ленты сдерживается низким временем доступа к данным.

Накопители на съемных магнитных дисках. Компания Iomega выпускает накопители ZIP и JAZ. ZIP-накопители работают с дисковыми носителями емкостью 100/250/750 Мб, по размеру незначительно превышающими стандартные гибкие диски, но несовместимыми, например, с ГМД 3,5 дюйма.

Совместимостью обладают устройства HiFD компании Sony. Они позволяют использовать как специальные носители емкостью 200 Мб, так и обычные гибкие диски. Распространение этих устройств сдерживается высокой ценой.

JAZ-носитель приближается к жестким дискам, но является сменным. В зависимости от модели накопителя емкость диска составляет 1 или 2 Гб.

Магнитооптические устройства обладают универсальностью. Предназначены для резервного копирования, обмена данными и их накопления. Высокая стоимость приводов и носителей. Параллельно развиваются 5,25- и 3,5-дюймовые накопители, носители которых отличаются в основном форм-фактором и емкостью. Ожидается увеличение емкости. Оптические диски CD и DVD состоят из множества ячеек, имеющих свои уникальные двоичные номера. Цифровая запись данных на дисках отличается высокой плотностью. Стандартный компакт-диск может хранить примерно 650 Мб данных. Большие объемы данных характерны для *мультимедийной информации* (графика, музыка, видео). На компакт-дисках распространяют программные *мультимедийные* продукты (книги, журналы, энциклопедии и т. п.). Однако на стандартных дисководах CD-ROM невозможно записывать данные, для этого используют дисководы CD-RW. Основным параметром дисководов CD-ROM является скорость чтения данных, измеряемая в кратных долях. За единицу измерения принята скорость чтения музыкальных компакт-дисков, составляющая в пересчете на данные 150 Кб/с. Дисковод с удвоенной скоростью чтения обеспечивает 300 Кб/с, с учетверенной — 600 Кб/с и т. д. В настоящее время устройства чтения CD-ROM имеют производительность 48x — 56x. Для заготовок, рассчитанных на однократную запись, скорость в соответствующих устройствах не уступает скорости чтения. Для многократной записи скорость записи может составлять 12x—24x (x = 150 Кбайт/с).

Флэш-диски — небольших размеров современное устройство хранения данных на основе энергонезависимой флэш-памяти. Подключается к разъему USB, допускает «горячее» подключение, распознается как жесткий диск, не требует установки драйвера. Емкость флэш-дисков — от 32 Мб до 1—4 Гб и выше.

Современная индустрия записи информации разного вида вызывает потребность в развитии новых носителей и форматов. Диски HD-DVD и BD обеспечивают одновременное чтение данных с диска и запись новой информации. Появляются сообщения о разработках новых форматов, обладающих огромной емкостью.

Например, интересный проект предложен Альянсом HVD, в который вошел ряд компаний. Это разработка голографических дисков Holographic Versatile Disk (HVD). Емкость такого диска — от 100 до 1000 Гб. Используется не один луч лазера, а сразу миллион. Скорость чтения при этом может достигать 1 Гбит/с.

Корпорация New Medium Enterprises собирается предоставить оптический четырехслойный диск Versatile Multilayer Disk (VMD), способный вместить 20 Гб и использующий обычный красный лазер.

Компания Iomega запатентовала технологию Articulated Optical-Digital Versatile Disc (AO-DVD), связанную с дальнейшим уменьшением размеров пита и появлением коротковолновых лазеров. В результате для хранения информации будут использоваться наноструктуры — участки, имеющие размеры меньше, чем длина волны лазера. Теоретически на такой диск можно будет записать до 800 Гб данных.

Компания D Data разрабатывает Digital Multilayer Disk (DMD) — диск для красного лазера, поддерживающий до шести слоев и имеющий емкость 15 Гб. Принцип его действия — активный слой под действием сфокусированного луча лазера начинает светиться (эффект флуоресценции). Количество слоев может быть шесть и более, главное, точно сфокусировать лазер на нужном слое.

Например, проект компании Colossal Storage Corporation связан с созданием 3,5-дюймовых дисков Atomic Holographic Disk емкостью 10 Тб.

Таким образом, ближайшее будущее будет занято дальнейшим развитием и улучшением оптических накопителей двух основных форматов — HD-DVD и BD. Появятся многослойные диски, вырастут скорости передачи данных, упадут цены. А потом придет время новых технологий.

2.7. Понятие системного и служебного (сервисного) программного обеспечения. Операционные системы

Назначение и структура системного программного обеспечения компьютера. В течение многих десятилетий создавались программы для работы аппаратной части компьютера (Hardware) и обработки данных на нем (Software). Совокупность необходимых программ для осуществления этих задач и составляет ПО ЭВМ.

Программное обеспечение ЭВМ имеет следующую структуру (рис. 2.3).

Характеристики составляющих элементов ПО, функции утилит, назначение, основные функции. *Базовое ПО* включает BIOS (Basic Input/Output System) — базовую систему ввода/вывода,

Рис. 2.3. Структура ПО ЭВМ

которая осуществляет первый этап загрузки операционной системы (ОС).

Тесты и BIOS входят в ПЗУ ПК, расположенного на системной плате, которое питается от элемента питания и поэтому не стирается при выключении ПК.

BIOS устанавливает связь между техническими средствами и ОС. Она выполняет наиболее простые и универсальные функции, связанные с вводом/выводом. Базовое ПО содержит также тест функционирования ПК, который проверяет работу памяти и устройств при включении питания, и программу вызова загрузчика ОС. BIOS является общей и неизменяемой частью всех ОС для данной модели ПК.

Системное ПО включает операционную систему (ОС), драйверы, оболочки, утилиты.

ОС — комплекс программ, управляющих всеми узлами ЭВМ и внешними устройствами и обеспечивающих максимальную производительность ЭВМ и системы в целом. ОС запускает про-

граммы, обеспечивает требуемый режим обработки и защиты данных, выполняет различные сервисные функции. Каждая программа пользуется услугами ОС и работает под ее управлением. Управляющие программы поддерживают автоматическую смену заданий, обеспечивая этим непрерывную работу ЭВМ без вмешательства человека.

Программы управления задачами считывают и приоритетно обрабатывают входные потоки задач, управляют параллельным выполнением заданий.

Программы управления данными предназначены для организации, идентификации, каталогизации, хранения и выборки обрабатываемых данных. Эти программы обеспечивают ввод-вывод данных разной структуры. Они планируют организацию размещения данных на внешних носителях и обеспечивают доступ к ним, распределяют оперативную память под программы, реализуют обмен данными между оперативной и внешней памятью, управляют внешними устройствами ввода-вывода.

Программы управления восстановлением регистрируют сбои в процессоре и внешних устройствах, ведут журнал учета сбоев, выявляют возможность завершения задачи после сбоя или переводят систему в состояние ожидания.

Программы конфигурации системы определяют характеристики конкретной реализации системы: тип и характеристики компьютера, имя, версию и редакцию ОС, местное (национальное) представление данных и т. п.

Поэтому выбор ОС очень важен.

ОС имеет следующие модули:

- командный процессор, который выполняет команды пользователя;
- средство управления файловой системой, которое осуществляет обмен файлами между устройствами;
- драйверы устройств — специальные программы, которые обеспечивают управление работой устройств и информационным обменом между ними;
- средство управления графическим интерфейсом с помощью мыши;
- некоторые сервисные программы (утилиты);
- справочную систему, которая служит для оперативного получения необходимой информации.

Драйверы — внешние программы, расширяющие возможности ОС, позволяют работать с выбранным внешним устройством.

Оболочки — программы, оптимизирующие работу ОС с помощью более удобного и наглядного способа общения с ПК, чем штатные средства ОС.

Утилиты — сервисные программы, позволяющие обслуживать диски (проверять, сжимать, дефрагментировать и т. д.), архивировать файлы, работать в компьютерных сетях, резервировать (копировать) данные.

Архиваторы позволяют сжимать (упаковывать) информацию на дисках, т. е. создавать копии файлов меньшего размера, а также объединять копии нескольких файлов в один архивный.

Антивирусы — это программы, предназначенные для сканирования и распознавания на компьютере пользователя программ или скриптов, макросов, которые могут причинить вред данным пользователя или существенно замедлить работу компьютера. Используются для лечения зараженных компьютерными вирусами (КВ) файлов. Антивирусные программы:

- полифаги — универсальные, проверяют файлы, сектора дисков и оперативную память и осуществляют поиск вирусов (Antiviral, Toolkit Pro);
- ревизоры — подсчитывают контрольные суммы длин файлов, даты последней модификации и т. п.;
- блокировщики — зашиты в Setup ПК — BIOS Setup, перехватывают вирусоопасные ситуации и сообщают об этом пользователю и др.

Должны использоваться самые последние версии (см. раздел 2.11).

Классификация операционных систем, базовые технологии работы в ОС. Для ПК IBM PC чаще всего используются следующие ОС:

- MS-DOS или совместимые с ней ОС PC DOS фирмы IBM и Novell DOS фирмы Novell;
- MS Windows разных версий, начиная с 3.x до XP Professional, Vista и Windows 7;
- OS/2 3.0 Warp фирмы IBM.

Для ПК, используемых в локальных сетях, используются следующие ОС:

- Netware разных версий фирмы Novell;
- Windows NT Server фирмы Microsoft;
- UNIX; OS/2.

Первая версия ОС для ПК IBM PC — MS-DOS 1.0 — появилась в 1981 г. Фирма IBM обратилась к Microsoft, которой была

известна ОС для ПК небольшой фирмы Seattle Computer Products. Билл Гейтс за 50 000 долл. приобрел права на эту OS, которая в дальнейшем послужила основой для MS-DOS. В 1981 г. первая версия PC/MS-DOS работала на IBM PC. В 1987 г. Microsoft разработала версию 3.3, которая стала стандартом на 3—4 года. Версия 6.0 содержит средства сжатия информации, антивирусные программы. Версия 6.2 надежна, устойчива. В каждой последующей версии много усовершенствований.

Основные составные части MS-DOS:

1. BIOS — система ввода/вывода.

2. Загрузчик ОС — считывает в память два модуля ОС, завершает ее загрузку.

3. Дисковые файлы: IO.SYS — дополняет BIOS и MSDOS.SYS — реализует основные функции DOS.

4. Командный процессор (COMMAND.COM) — обрабатывает команды, вводимые пользователем. Внутренние команды: `Type`, `Dir`, `Copy`.

5. Внешние команды — программы, поставляемые вместе с ОС.

6. Драйверы устройств — специальные программы, дополняют BIOS и обеспечивают обслуживание новых или нестандартных устройств.

Оболочка ОС — программа для наглядного и простого выполнения основных команд с использованием меню, защитой от необдуманных действий и контекстной помощью. Она входит в комплект утилит и предназначена для облегчения взаимодействия пользователя с компьютером. Например, оболочка Norton Commander — пользовательский интерфейс, состоящий из двух панелей, на каждой из которых можно отображать дерево каталогов, каталоги и находящиеся в них файлы, служебную информацию. Оболочка Windows — графический многооконный интерфейс.

2.8. Файловая структура ОС. Операции с файлами

При хранении данных решаются две проблемы: как сохранить данные в компактном виде и как обеспечить быстрый и удобный доступ к ним. В качестве единицы хранения данных принят объект переменной длины, называемый файлом. *Файл* — это последовательность произвольного числа байтов, обладаю-

щая уникальным собственным именем. Обычно в отдельном файле хранят данные, относящиеся к одному типу. В этом случае тип данных определяет тип файла.

Все программы и данные хранятся во внешней памяти ПК в виде файлов. Файл — определенное количество информации (программ или данных), имеющее имя и хранящееся в памяти. Имя файла фактически должно нести в себе адресные данные, помогающие обеспечить доступ к данным, хранящимся в файле. Полное имя файла состоит из двух частей, разделенных точкой: собственно имени и расширения. Имя дает пользователь, а расширение задается автоматически.

Например, расширения для файлов: исполняемых — .exe, .com; текстовых —.txt, .doc; графических — .bmp, .gif, .jpg; звуковых — .wav, .mid; видео — .evi; программных — .bas, .pas, .asm и др.

На каждом диске может храниться большое количество файлов. *Файловая структура* — иерархическая структура хранения файлов и организации каталогов. Каждый диск разбивается на две области: область хранения файлов и каталог, который содержит название файла и указание на начало его размещения на диске. Каталог — это оглавление. Каталог (директория, папка) — специальное место на диске, в котором хранятся имена файлов, их размеры, время обновления, свойства (рис. 2.4).

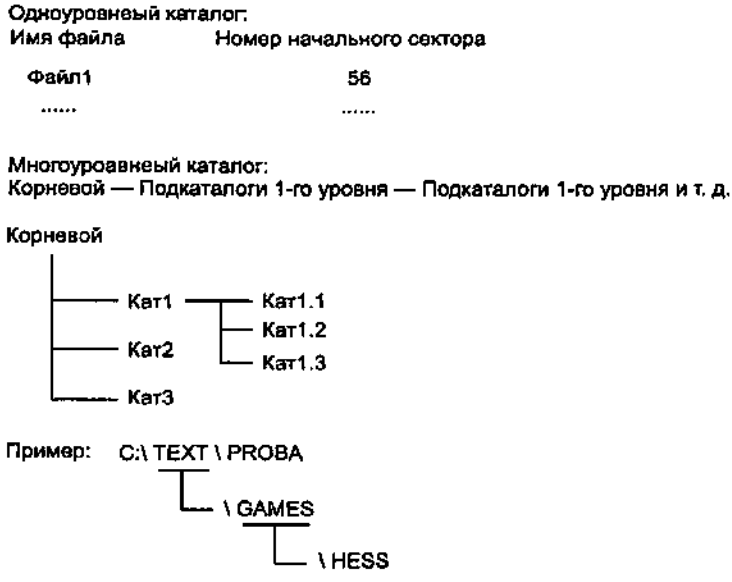


Рис. 2.4. Пример файловой структуры

В качестве вершины структуры служит имя носителя, на котором сохраняются файлы.

Далее файлы группируются в каталоги (папки), внутри которых могут быть созданы вложенные каталоги (подпапки). Путь доступа к файлу начинается с имени устройства и включает все имена каталогов, через которые проходит. В качестве разделителя используется символ \ (обратная косая черта, обратный слэш). Путь — последовательность из имен каталогов или символов, разделенных символом \. Путь задает маршрут от текущего или корневого каталога к каталогу, в котором находится нужный файл. Например, C:\Text\Games\proba.txt — путь, полное имя файла (собственное имя файла с путем доступа к нему).

Все современные дисковые ОС предусматривают создание файловой системы, предназначенной для хранения данных на дисках и обеспечения доступа к ним. Принцип организации файловой системы — *табличный.*

Поверхность жесткого диска рассматривается как трехмерная матрица, измерениями которой являются номера *поверхности, цилиндра* и *сектора*. Под цилиндром понимается совокупность всех дорожек, принадлежащих разным поверхностям и находящихся на равном удалении от оси вращения. Данные о том, в каком месте диска записан файл, хранятся в системной области диска. Формат служебных данных определяется конкретной файловой системой. Нарушение целостности служебных данных приводит к невозможности воспользоваться данными, записанными на диске. К системной области предъявляются особые требования по надежности. Целостность, непротиворечивость и надежность этих данных регулярно контролируется средствами операционной системы. Наименьшей физической единицей хранения данных является *сектор.* Размер сектора равен 512 байт. Теоретически возможна самостоятельная адресация для каждого сектора. Но для дисков большого объема такой подход неэффективен, поэтому группы секторов объединяются в кластеры. *Кластер* является наименьшей единицей адресации при обращении к данным. Размер кластера, в отличие от размера сектора, строго не фиксирован. Обычно он зависит от емкости диска.

Операционные системы MS-DOS, OS/2, Windows 95 и др. используют файловую структуру на основе таблиц размещения файлов (FAT-таблицы), состоящих из 16-разрядных полей. Такая файловая система называется *FAT16.* Она позволяет разместить в FAT-таблицах не более 65 536 записей (2^{16}) о местоположении

единиц хранения данных. Для дисков объемом от 1 до 2 Гб длина кластера составляет 32 Кб (64 сектора). Это не вполне рациональный расход рабочего пространства, поскольку даже маленький файл полностью оккупирует весь кластер. Для дисков объемом примерно 2 Гб потери, связанные с неэффективностью этой файловой системы, могут составлять от 25 до 40 % полной емкости диска.

Начиная с ОС Windows 98, семейство Windows (Windows Me/2000/XP) поддерживает более совершенную версию файловой системы на основе FAT-таблиц — *FAT32* с 32-разрядными полями в таблице размещения файлов. Для дисков размером до 8 Гб эта система обеспечивает размер кластера 4 Кб (8 секторов).

Операционные системы Windows NT и Windows XP способны поддерживать совершенно другую файловую систему — *NTFS*. В ней хранение файлов организовано иначе — служебная информация хранится в главной таблице файлов (*MFT*). В системе NTFS размер кластера не зависит от размера диска, и потенциально для очень больших дисков эта система должна работать эффективнее, чем FAT32.

Обслуживание файловой структуры. Несмотря на то что данные о местоположении файлов хранятся в табличной структуре, пользователю они представляются в виде иерархической структуры, так удобнее, а все необходимые преобразования берет на себя ОС. К функции обслуживания файловой структуры относятся следующие операции, происходящие под управлением ОС:

- создание файлов и присвоение им имен;
- создание каталогов (папок) и присвоение им имен;
- переименование файлов и каталогов (папок);
- копирование и перемещение файлов между дисками компьютера и между каталогами (папками) одного диска;
- удаление файлов и каталогов (папок);
- навигация по файловой структуре с целью доступа к заданному файлу, каталогу (папке);
- управление атрибутами файлов.

Создание и именование файлов. Поскольку из определения файла вытекает, что он может иметь нулевую длину, то фактически создание файла состоит в присвоении ему имени и регистрации его в файловой системе. Это одна из функций ОС.

По способам именования файлов различают короткое и длинное имена. Согласно принятому в MS-DOS соглашению (соглашение 8.3 для IBM PC) на собственно имя файла отводит-

ся 8 символов, на расширение — 3. Как имя, так и расширение могут включать только алфавитно-цифровые символы латинского алфавита. Имеются исключения: можно использовать специальные символы (восклицательный знак, символ подчеркивания, дефис, тильде и т. п.), а некоторые версии MS-DOS даже допускают использование в именах файлов символов русского и других алфавитов. Сегодня имена файлов, записанные в соответствии с соглашением 8.3, считаются короткими. Недостатком коротких имен является их низкая информативность.

С появлением ОС Windows 95 было введено понятие «длинное имя» (до 256 символов), чего вполне достаточно для создания содержательных имен файлов. Длинное имя может иметь любые символы, кроме девяти специальных: / (слэш), \ (обратный слэш), : (двоеточие), * (звездочка), ? (вопросительный знак), " (кавычки), < (меньше), > (больше), | (вертикальная черта). В имени можно использовать пробелы и точки. Расширением имени считаются все символы, идущие после последней точки, их может быть и больше трех. Введение длинных имен потребовало внесения изменений в организацию файловых систем на основе FAT. Появился термин VFAT, обозначающий файловую систему на основе FAT с поддержкой длинных имен. Файловая система NTFS поддерживает длинные имена изначально. Использование длинных имен файлов в ОС семейства Windows имеет ряд особенностей.

1. Если длинное имя файла включает пробелы, то в служебных операциях его надо заключать в кавычки. Рекомендуется не использовать пробелы, а заменять их символами подчеркивания.

2. В корневой папке диска нежелательно хранить файлы с длинными именами. В файловых системах на основе FAT количество единиц хранения в этой папке ограниченно. Чем длиннее имена, тем меньше файлов можно разместить в корневой папке.

3. Кроме ограничения на длину имени файла (256 символов) существует гораздо более жесткое ограничение на длину *полного имени файла*. Полное имя не может быть длиннее 260 символов.

4. В длинных именах файлов разрешается использовать символы любых алфавитов, но, если документ готовится для передачи, с заказчиком необходимо согласовать возможность воспроизведения файлов с такими именами на его оборудовании.

5. Прописные и строчные буквы в именах не различаются ОС.

6. Расширение имени файла используется для передачи операционной системе, исполняющей программе или пользователю информации о том, к какому типу относятся данные, содержа-

Управление атрибутами файлов. Кроме имени и расширения ОС хранит для каждого файла дату его создания (изменения) и несколько флаговых величин, называемых *атрибутами файла*. Атрибуты — это дополнительные параметры, определяющие свойства файлов. ОС позволяет их контролировать и изменять. Состояние атрибутов учитывается при проведении автоматических операций с файлами.

Основных атрибутов четыре: **Только для чтения (Read only)**; **Скрытый (Hidden)**; **Системный (System)**; **Архивный (Archive)**.

Атрибут **Только для чтения** ограничивает возможности работы с файлом. Его установка означает, что файл не предназначен для внесения изменений.

Атрибут **Скрытый** сигнализирует операционной системе о том, что данный файл не следует отображать на экране при проведении файловых операций. Это мера защиты против случайного (умышленного или неумышленного) повреждения файла.

Атрибутом **Системный** помечаются файлы, обладающие важными функциями для работы самой операционной системы. Их отличительная особенность в том, что средствами ОС их изменить нельзя.

Атрибут **Архивный** в прошлом использовался для работы программ резервного копирования. Предполагалось, что любая программа, изменяющая файл, должна автоматически устанавливать этот атрибут, а средство резервного копирования должно его сбрасывать. Таким образом, очередному резервному копированию подлежали только те файлы, у которых этот атрибут был установлен. Современные программы резервного копирования используют другие средства для установления факта изменения файла, и данный атрибут во внимание не принимается, а его изменение вручную средствами ОС не имеет практического значения.

2.9. ОС MS-DOS и Windows. Работа с файлами и каталогами

Операционная система MS-DOS определяет общие правила запуска программ, управления данными и доступа к ресурсам компьютера.

Начальная загрузка MS-DOS выполняется автоматически при включении электропитания ПК или при нажатии кнопки **Reset** на корпусе компьютера. Для выполнения начальной загрузки необ-

ходимо, чтобы ОС была записана на логическом диске (части жесткого диска) C: или чтобы системная дискета с ОС находилась в дисководе A:. Программа-загрузчик, находящаяся в начальном секторе логического диска или дискеты, считывает в память ПК основные файлы — IO.SYS и MSDOS.SYS. На экране появляется сообщение «Starting MS-DOS...» (Загрузка MS-DOS...).

Из корневого каталога загрузочного диска читается файл конфигурации системы CONFIG.SYS, с помощью команд которого загружаются драйверы устройств и устанавливаются параметры ОС. Из корневого каталога загрузочного диска или файла CONFIG.SYS читается командный процессор COMMAND.COM, которому передается управление.

Командный процессор выполняет командный файл AUTOEXEC.BAT. Массив файла AUTOEXEC.BAT содержит команды и программы, выполняемые при каждом запуске ПК. Для одновременного с загрузкой появления нужной вам программы следует в конец файла AUTOEXEC.BAT написать команду ее вызова, например для вызова Norton Commander. Чтобы часто выполняемые программы запускать из любого каталога, их следует перечислить в команде `PATH`, например: `PATH C:\ USER` и т. д.

После загрузки ОС на экран выводится приглашение к вводу команд. Приглашение DOS обычно содержит информацию о текущем дисководе и каталоге, например: C:\>EXE. Для вызова встроенного справочника следует ввести команду `HELP`.

Файловая система — система управления данными. Она поддерживает дисководы, обозначаемые латинской буквой и двоеточием (например: A:, B:, C:), систему каталогов, файлы.

Если требуется выполнить программу или команду DOS, необходимо набрать эту команду на клавиатуре и нажать клавишу **Enter**. Чтобы грамотно работать, необходимо знать команды MS-DOS, которые можно разделить на следующие группы:

- для работы с дисками;
- для работы с файлами;
- для работы с каталогами;
- для управления системой.

Команды имеют следующую структуру:

<имя команды> [<список параметров>] [<список ключей>]

Параметры указывают на объекты, над которыми выполняют операции, ключи уточняют действие команды. Рассмотрим, на-

пример, команду DIR. Если не указать параметры и ключи, она выведет на экран имена файлов (каталогов) и их разные признаки (размер, дату создания и др.). Команда DIR *.TXT выведет на экран список файлов с расширением .txt. Самый полный синтаксис команды должен быть такой:

```
DIR [диск:] [путь] [имя файла] [/ ключ]
```

Ввод команды завершается нажатием клавиши **Enter**.

Команды работы с дисками.

VERIFY — установить или отменить режим проверки правильности записи на диск.

VOL — вывод метки диска.

ASSIGN — назначить дисководу другое логическое имя (букву).

CHKDISK — проверка диска на правильность файловой системы.

DISKCOMP — сравнение дискет.

DISKCOPY — копирование дискет.

FDISK — разметка жесткого диска.

FORMAT — форматирование (инициализация) диска.

JOIN — логически присоединить дисковод к указанному каталогу.

LABEL — узнать или установить метку диска.

Команды работы с файлами.

COPY — копирование файлов.

DEL — удаление файлов.

ECHO — выдать сообщение из пакетного командного файла.

FOR — организация циклов.

GOTO — переход на метку в пакетном командном файле.

IF — проверка условия в пакетном командном файле.

PAUSE — приостановить выполнение пакетного командного файла.

REM — комментарий в пакетном командном файле.

REN — изменить имя файла.

SHIFT — сдвиг номеров параметров пакетного командного файла.

TYPE — просмотр файла (вывод файла на экран).

ATTRIB — установить или показать атрибуты файла.

BACKUP — создать архивные копии файлов.

DEBUG — просмотр, изменение, дизассемблирование файлов.

EDLIN — примитивный редактор текстов.

EXE2BIN — преобразование EXE-файла в двоичный код.

FASTOPEN — ускорение открытия файлов.

FC — сравнение файлов.

FIND — поиск подстроки в файлах.

MORE — постраничная выдача на экран дисплея.

PRINT — распечатка на принтере текстовых файлов в фоновом режиме.

RECOVER — восстановить файл, содержащий сбойные участки.

REPLACE — заменить файлы их новыми версиями.

RESTORE — прочитать файлы из архива, созданного с помощью команды BACKUP.

SHARE — установить многопользовательский режим использования файлов.

SORT — сортировка данных.

XCOPY — копирование файлов (имеет больше возможностей, чем COPY).

Команды работы с каталогами.

CD — сменить текущий каталог или показать имя текущего каталога.

DIR — выдать список файлов в каталоге.

MD — создать новый каталог.

PATH — установить список каталогов для поиска команд.

RD — удалить каталог.

APPEND — задать дополнительные каталоги для поиска данных.

SUBST — заменить имя каталога обозначением дисковода.

TREE — вывести структуру каталогов на диске.

Команды управления системой.

BREAK — установить режим проверки ввода комбинации клавиш **Ctrl+C**.

CLS — очистить экран дисплея.

CTTY — сменить устройство ввода-вывода для команд MS-DOS.

DATE — получить или изменить текущую дату.

EXIT — окончить работу командного процессора COMMAND.COM.

PROMT — установить вид приглашения MS-DOS.

SET — установить переменную окружения.

TIME — получить или установить текущее время.

VER — выдать номер версии MS-DOS.

COMMAND — запустить командный процессор MS-DOS.

GRAFTABL — загрузка таблицы шрифтов символов с кодами 128—255.

GRAPHICS — подготовка к печати графической копии экрана.

LINK — редактор связей.

MODE — установить режимы работы устройств.

SYS — скопировать системные файлы на диск.

Опробование утилит DOS можно выполнить с помощью программы NCD из комплекса Norton Utilities, позволяющей выводить на экран дерево каталогов, осуществлять вход в определенный каталог, переход в другой каталог и т. д.

Форматирование (инициализация) дискет осуществляется для подготовки их к работе. Программа FORMAT предназначена для подготовки системной дискеты и для очистки дискеты от данных с отметкой всех ее дефектных участков. Форма записи команды форматирования:

```
FORMAT <дисковод: (параметры)>
```

Например: `FORMAT A:/U` — безусловное форматирование дискеты A:.

Norton Commander является DOS-программой, поэтому она не понимает длинные имена файлов, можно использовать не более 8 символов для имени и не более 3 для расширения.

Запуск Norton Commander осуществляется набором команды NC. Выход из Norton Commander выполняется нажатием клавиши **F10**.

Правая и левая панели — большие синие прямоугольники, на которых представлены каталоги (директории) дисков, одна из панелей (на которой находится указатель) является активной. Под панелями находится командная строка с приглашением ввести команду или сформировать ее, используя имена файлов на панелях. Под командной строкой имеется строка подсказки — функциональные клавиши **F1—F10** с обозначением основных команд:

- **F1 (Help)** — вызов контекстной подсказки;
- **F2 (Menu)** — вызов пользовательского меню;

- **F3 (View)** — вызов указанного файла для просмотра в режиме текста или кода;
- **F4 (Edit)** — вызов простейшего редактора для редактирования указанного текстового файла;
- **F5 (Copy)** — копирование указанного файла или каталога (или отмеченная их группа) с активной панели на диск, отражаемый пассивной панелью;
- **F6 (RenMov)** — переименование или перемещение файлов или директорий с активной панели на пассивную;
- **F7 (MkDir)** — создание каталога на активной панели;
- **F8 (Delete)** — удаление указанных или отмеченных файлов или каталогов;
- **F9 (PullDn)** — вызов горизонтального меню в верхней строке экрана. **Esc** — выход из меню;
- **F10 (Quit)** — выход из Norton Commander.

Выделить файлы в Norton Commander можно с помощью клавиши **Insert** (выделение одного файла) или сочетания клавиш **Enter**++ (выделение всех файлов в каталоге).

Клавиши для управления панелями:

- **Tab** — сделать активной другую панель;
- **Ctrl+O** — убрать или вывести панели;
- **Ctrl+P** — убрать неактивную панель или вывести на экран;
- **Ctrl+U** — поменять панели местами;
- **Ctrl+F1** — убрать левую панель или вывести ее на экран;
- **Ctrl+F2** — убрать правую панель или вывести ее на экран;
- **Ctrl+F10** — разрезание и слияние файлов;
- **Alt+F1** — вывести в левой панели оглавление дисков;
- **Alt+F2** — вывести в правой панели оглавление дисков;
- **Alt+F5** — сжатие выделенных файлов (помещение в архив);
- **Alt+F6** — извлечение файлов из выделенных архивов.

Для изменения типа панели можно использовать команды меню:

- **Brief** — краткий формат оглавления каталогов;
- **Full** — полный формат оглавления каталогов;
- **Tree** — дерево каталогов;
- **Info** — состояние;
- **Directory Information** — паспорт каталога;
- **View** — просмотр;
- **Find File Panel** — панель поиска;
- **Link** — связь.

В связи с тем что на современных компьютерах, как правило, отсутствует Norton Commander, практическую работу по созданию файла можно выполнить, вызвав командную строку MS-DOS из главного меню ОС Windows с помощью команд **Пуск → Программы → Стандартные → Командная строка**.

Для создания небольшого собственного файла в командной строке следует набрать команду `COPY CON <имя файла>`, т. е. копирование файла с консоли, и нажать на клавишу **Enter**. Далее ввести строки данных. Завершить ввод данных в файл следует нажатием клавиш **Ctrl+Z** или **F6**.

Работа с файлами в ОС Windows. При изучении пользовательского интерфейса ОС Windows следует руководствоваться указаниями, которые можно получить с помощью команды главного меню **Справка**. Необходимую информацию можно вывести на экран, вводя в текстовое поле предметного указателя ключевое слово. Например, введя в текстовое поле слово «Ярлыки» и нажав кнопку **Вывести**, получите информацию о создании и удалении ярлыков.

Работу с файловой структурой ОС Windows выполняют, как правило, используя программу Проводник.

Запустить программу Проводник можно с помощью:

- команд меню **Пуск → Программы → Проводник**;
- контекстного меню ярлыка **Мой компьютер**;
- контекстного меню кнопки **Пуск**;
- двойного щелчка левой кнопкой мыши по созданному ярлыку программы.

Для создания ярлыка программы Проводник необходимо после нажатия кнопки **Пуск** ввести команды **Поиск → Файлы и папки**. Далее в диалоговом окне **Найти: Все файлы** в поле **Имя** ввести слово «Проводник» и нажать кнопку **Найти**. С помощью команды контекстного меню **Копировать** ярлык программы Проводник заносят в буфер обмена, а затем с помощью команды **Вставить** помещают на рабочий стол.

Файлы следует создавать с помощью команд **Файл → Создать → Новый документ**. Переименовать их можно с помощью команды контекстного меню.

При изучении программы Калькулятор (стандартный и инженерный) следует, используя произвольно выбранные числа, выполнить арифметические действия и нахождение основных функций. Функцию каждой кнопки можно изучить, щелкнув по

ней правой кнопкой мыши. Появится вопрос «Что это такое?». После второго щелчка появляется объяснение.

Для завершения работы на компьютере закрывают все файлы и папки, а затем, нажав кнопку **Пуск** → **Завершение работы**, выключают его только после соответствующего объявления на экране дисплея.

Контрольные вопросы

1. Каковы назначение и основные функции операционных систем?
2. Каковы основные группы команд операционной системы MS-DOS и их перечень?
3. Что такое файл?
4. Что такое каталог (директория) и для чего он служит?
5. В чем состоит назначение программы-оболочки ОС?
6. Каковы основные элементы панелей файлового менеджера Norton Commander?
7. Какие пункты меню имеются в Norton Commander?
8. Перечислите основные команды MS-DOS.
9. Как создать ярлык и какие операции можно с ним выполнить?
10. Как выполняется работа с файловой структурой ОС Windows?

2.10. Установка и обновление программного обеспечения

Выполнение установки ПО компьютера начинается с процесса его начальной загрузки, т. е. при включении ПК тестируются его устройства и начинается загрузка операционной системы. Начальная загрузка осуществляется автоматически при включении электропитания ПК или при нажатии на его корпусе кнопки **Reset**. Начальная загрузка (перезагрузка) может быть выполнена пользователем при одновременном нажатии на клавиатуре клавиш **Ctrl**, **Alt**, **Delete**. При начальной загрузке автоматически запускаются находящиеся в постоянной памяти компьютера (BIOS) программы: проверки оборудования и конфигурирования компьютера.

После проверки оборудования и инициализации устройств ПК программа начальной загрузки пытается загрузить ОС с сис-

темной дискеты на дисководе A:. Если она обнаруживает такую дискету, то пытается прочесть с ее начального сектора программу-загрузчик ОС. Если загрузка выполняется с жесткого диска, т. е. на логическом диске (части жесткого диска) была записана ОС, то следует нажать любую буквенно-цифровую клавишу, пробел или **Enter** для продолжения загрузки. О загрузке см. раздел 2.9.

Если MS-DOS готова к работе, то на экран выводится приглашение для ввода команд, например `C:\>`. Для вызова на экран сразу своей программы следует в конец файла AUTOEXEC.BAT вставить соответствующую команду (например, вызова оболочки или другой ОС).

При загрузке ОС Windows считывает свою конфигурацию из основного каталога файлов инициализации WIN.INI (содержит параметры работы ОС) и SYSTEM.INI (содержит настройки ОС на внешнее окружение — аппаратные средства ПК и параметры DOS). По умолчанию Windows запускается в расширенном режиме, а для запуска в стандартном режиме следует задать еще параметр `/S`. Если функционирование системы затруднено, то следует использовать определенные параметры.

Для обновления ОС или установки последних версий (например, Windows XP) следует из главного меню вызвать Центр справки и поддержки, в поле **Найти** ввести ключевое слово поиска (например, «обновление») и далее выполнять действия, следуя указаниям, появляющимся в диалоговом окне (рис. 2.5).

Установка программных оболочек. Программную оболочку ОС MS-DOS можно ввиду ее небольшой конфигурации запускать с дискеты, а можно, если она имеется на жестком диске, из файла AUTOEXEC.BAT, в котором должен быть указан путь и команда `NC`.

Оболочка Windows — программа, которая запускается первой и запускает остальные программы ОС Windows. Если она имеется на жестком диске, то может быть автоматически запущена из файла AUTOEXEC.BAT. Для запуска Windows из командной строки MS-DOS нужно задать команду `WIN`. На экран будет выведена заставка, а затем рабочий стол и другие компоненты интерфейса (оболочки) ОС. Если система отсутствует на диске, то ее следует установить с дистрибутивных дисков или пакета системных дискет, вставляя их в дисковод A: и следуя появляющимся указаниям.

Для установки в среде Windows XP современного прикладного программного обеспечения и обновления существующей вер-

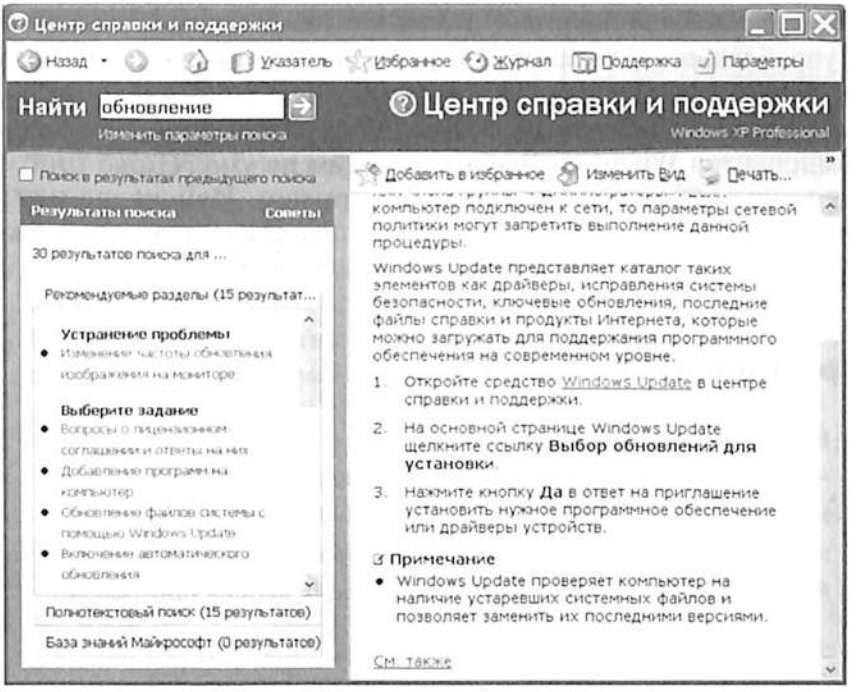


Рис. 2.5. Окно Центра справки и поддержки

Рис. 2.6. Окно установки и удаления программ

сии программ следует сначала выполнить команды **Пуск → Панель управления → Установка и удаление программ**. В диалоговом окне **Установка и удаление программ**, слева, имеются вкладки: **Изменение или удаление программ**; **Установка программ**; **Установка компонентов Windows**; **Выбор программ по умолчанию** (рис. 2.6). Выберите необходимую вкладку (например, **Изменение или удаление программ**), а затем нажмите кнопку **Изменить** или **Удалить**. Далее следуйте инструкциям, появляющимся на экране.

Контрольные вопросы

1. Как установить ОС последних версий?
2. Как установить программную оболочку ОС MS-DOS?
3. Как установить программную оболочку ОС Windows?
4. Как установить современное прикладное программное обеспечение?
5. Как выполнить обновление существующих версий программ?

2.11. Работа с программами-архиваторами. Антивирусные программы

Программы-архиваторы (упаковщики) — это программы-утилиты, предназначенные для сжатия информации на дисках. С их помощью можно создавать копии файлов меньших размеров, а также объединять копии нескольких файлов в один архивный файл.

Архивный файл (архив) — набор сжатых файлов (файл), помещенных в единый файл. Он имеет оглавление содержащихся в нем файлов, включающее имя файла, сведения о каталоге, где он находится, дату изменения файла, его размер и код контроля для проверки целостности архива.

Наиболее распространенными для MS-DOS являются программы PKZIP/PKUNZIP, которые фактически стали стандартом сжатия файлов. Используют также ARJ, PAK, PKPAK и др.

Для работы с архивом ZIP-файлов пакет может содержать следующие программы:

- PKZIP — помещение файлов в архив;
- PKUNZIP — извлечение файлов из архива;

- PKZIPFIX — восстановление поврежденного архивного файла;
- ZIP2EXE — создание самораспаковывающихся архивных файлов;
- PKZFIND — поиск файла на диске и в ZIP-архивах.

Поместить файлы в архив можно, указав один из трех режимов:

- **Add** — добавление в архив всех файлов (по умолчанию, например, задав команду `PKZIP mydoc`);
- **Update** — добавление в архив новых файлов (команда `U`, например, задав команду `PKZIP -u a: mydoc`, мы добавим файлы из текущего каталога);
- **Freshen** — добавление новых версий имеющихся в архиве файлов (команда `F`, например, задав команду `PKZIP -f a: mydoc b:\*.*`, мы добавим в наш архив новые версии файлов из корневого каталога диска B:).

Архивацию файлов из подкаталогов можно выполнить с помощью режима **гр**. Например, задав команду `PKZIP -rp a: mydoc c:\doc\*.*`, мы архивируем в mydoc все файлы из подкаталога doc на диске C:.

Для просмотра оглавления архива используется команда `PKUNZIP -v <имя архива (имена файлов)...>`.

Для создания архива в среде Windows 98 следует запустить программу Windows Commander и выполнить команды меню **Файл → Упаковать** (используется архиватор arj). Просмотрите содержимое архива. Сравните размеры исходных и архивных файлов. Распакуйте файлы с помощью команд **Файл → Распаковать**.

При создании архива в среде Windows XP следует последовательно задать команды **Пуск → Программы → Стандартные → Служебные → Архивация данных**. На экране появится диалоговое окно **Мастер архивации и восстановления**. Нажимая кнопку **Далее**, нужно следовать предлагаемым указаниям.

В настоящее время для ОС Windows (32- и 64-разрядных), Pocket PC, Linux, FreeBSD, Mac OS X, MS-DOS и OS/2 часто используют файловый архиватор WinRAR (рис. 2.7) с высокой степенью сжатия. Он является одним из лучших архиваторов по соотношению степени сжатия к скорости работы. Основные возможности WinRAR:

- создание архивов в форматах RAR и ZIP;
- распаковка файлов многих форматов;

- возможность шифрования архивов;
- возможность работы с файлами размером до 8,589 трлн (10^{12}) Гб;
- создание самораспаковывающихся, непрерывных и многотомных архивов;
- добавление в архивы дополнительной информации для восстановления архива в случае его повреждения;
- полная поддержка файловой системы NTFS и имен файлов в Unicode;
- поддержка командной строки.

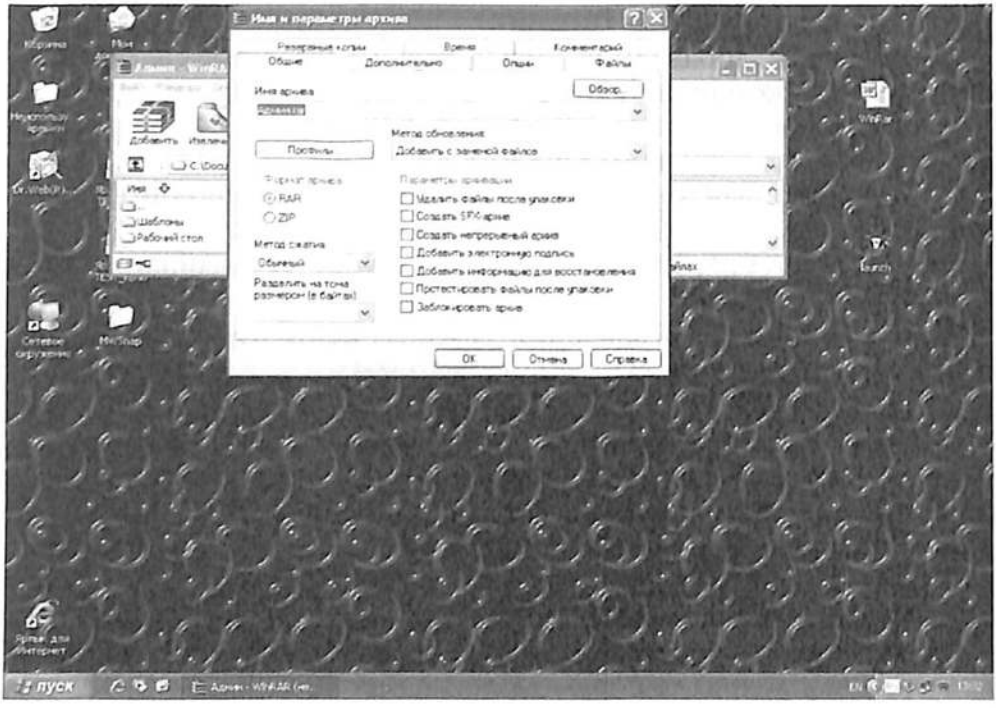


Рис. 2.7. Рабочее окно архиватора WinRAR

С версии 3.60 алгоритмом компрессии поддерживаются многоядерные процессоры и процессоры с технологиями Hyper-threading — это обеспечивает существенный прирост скорости сжатия. С версии 3.70 архиватор официально совместим с Windows Vista, создает SFX-архивы с запросом привилегий. С версии 3.80 поддерживаются архивы ZIP, которые содержат имена файла Unicode в формате UTF-8. С версии 3.90 появляется раз-

деление на 32- и 64-битные версии программы. 64-битная версия работает только на 64-битных ОС Windows и дает некоторый прирост производительности.

Компьютерные вирусы и антивирусные программы. *Компьютерные вирусы* — саморазмножающиеся (самокопирующиеся) программы, внедряющиеся в файлы, загрузочные сектора дисков и документы и наносящие им ущерб или уничтожающие их. К настоящему времени зарегистрированы тысячи компьютерных вирусов.

Компьютерные вирусы можно классифицировать по степени опасности, среде обитания, способу заражения и алгоритму функционирования.

По *степени опасности* компьютерные вирусы могут быть безвредными (уменьшают только свободную память из-за размножения), неопасными (добавляют разные эффекты), опасными (приводят к сбоям в работе компьютера), очень опасными (приводят к потере программ и данных, форматированию жестких дисков и т. д.).

По *среде обитания* компьютерные вирусы делят на файловые (внедряются в исполняемые файлы и активизируются при их запуске), загрузочные (заносятся в загрузочный сектор диска, при загрузке ОС внедряются в оперативную память), макровирусы (заражают файлы документов Word и Excel) и сетевые (любые вирусы, распространяющиеся по сети, часто они внедряются в компьютеры через зараженные файлы с серверов файловых архивов, по электронной почте как вложенный в сообщение файл или через Всемирную паутину в виде активных элементов — программ на языке Java или Visual Basic).

По *способу заражения среды обитания* компьютерные вирусы могут быть резидентными, когда после активации они попадают из среды обитания в оперативную память ЭВМ и остаются там, и нерезидентными, т. е. попадают в ОП ЭВМ только на время активации.

По *алгоритму функционирования* вирусы делят на:

- студенческие — низкой квалификации, легко обнаруживаются;
- паразитические — простые, изменяют содержимое файлов и секторов диска, их легко обнаружить и обезвредить;
- стелс — невидимки, маскирующие свое присутствие под программы ОС, их трудно обнаружить и обезвредить;

- черви (репликаторы) — распространяются по компьютерным сетям, вычисляют адреса компьютеров и записывают по ним свои копии;
- спутники — создают копии исполняемых файлов и др.;
- мутанты (призраки) — содержат алгоритмы шифровки/расшифровки; очень трудно обнаружить;
- троянские программы (квазивирусы) — маскируются под полезную программу, разрушают загрузочный сектор и файловую систему дисков, очень опасны.

Любой вирус независимо от класса имеет три блока: заражения, маскировки, выполнения деструктивных действий.

Для защиты от вирусов следует установить на ПК пакет антивирусных программ. Компьютер на наличие вирусов следует проверять ежедневно. Запустив антивирусную программу, следует прежде всего проверить оперативную память, чтобы предотвратить дальнейшее распространение вируса. Если антивирусная программа не удалит вирусы, то ПК нужно перезагрузить с незараженной и защищенной дискеты.

При выборе антивирусной программы необходимо учитывать не только процент обнаружения вирусов, но и способность обнаруживать новые вирусы, количество вирусов в антивирусной базе, частоту ее обновления, наличие дополнительных функций.

В настоящее время серьезный антивирус должен уметь распознавать не менее 25 000 вирусов. Зараженные вирусом файлы нужно либо удалить, либо вылечить.

Антивирусные программы делятся на детекторы, доктора, ревизоры, фильтры, вакцины.

Программы-детекторы (различают универсальные и специализированные) обеспечивают поиск и обнаружение вирусов в оперативной памяти и на внешних носителях и при обнаружении выдают соответствующее сообщение. Универсальные детекторы в своей работе используют проверку неизменности файлов путем подсчета и сравнения с эталоном контрольной суммы, но они не могут определить причины искажения файлов.

Специализированные детекторы выполняют поиск известных вирусов по их сигнатуре (повторяющемуся участку кода), но они не могут обнаруживать все известные вирусы.

Детектор, позволяющий обнаруживать несколько вирусов, называют полидетектором. Но и они могут находить только те вирусы, которые известны разработчикам таких программ.

Программы-доктора (фаги) находят зараженные вирусами файлы и лечат их, т. е. удаляют из файла тело программы вируса. Сначала фаги ищут вирусы в оперативной памяти, уничтожают их, а затем переходят к лечению файлов. Полифаги предназначены для поиска и уничтожения большого количества вирусов.

Программы-ревизоры — самые надежные средства защиты от вирусов. Ревизоры запоминают исходное состояние программ, каталогов и системных областей диска тогда, когда компьютер не заражен вирусом, а затем периодически или по желанию пользователя сравнивают текущее состояние с исходным. Обнаруженные изменения выводятся на экран. Как правило, сравнение состояний производят сразу после загрузки операционной системы. При сравнении проверяются длина файла, код циклического контроля (контрольная сумма файла), дата и время модификации, другие параметры.

Программы-фильтры (сторожа) — небольшие резидентные программы, предназначенные для обнаружения подозрительных действий при работе компьютера. Это следующие действия:

- попытки коррекции файлов с расширениями .com и .exe;
- изменение атрибутов файлов;
- прямая запись на диск по абсолютному адресу;
- запись в загрузочные сектора диска;
- загрузка резидентной программы.

При попытке какой-либо программы произвести указанные действия «сторож» посылает пользователю сообщение и предлагает запретить или разрешить соответствующее действие. Фильтры способны обнаружить вирус на самой ранней стадии его существования до размножения, но они не лечат файлы и диски. Для уничтожения вирусов требуется применить другие программы, например фаги.

Программы-вакцины (иммунизаторы) — резидентные программы, предотвращающие заражение файлов. Их применяют при отсутствии лечащих программ. Вакцинация возможна только от известных вирусов. Существенным недостатком таких программ являются их ограниченные возможности по предотвращению заражения от большого числа разнообразных вирусов.

Существует множество антивирусных программ. Один из наиболее известных и популярных антивирусов — *Norton AntiVirus* (производитель Symantec). Процент распознавания вирусов близок к 100 %. В программе используется механизм, который позво-

ляет распознавать новые неизвестные вирусы. В интерфейсе программы имеется функция LiveUpdate, позволяющая одним щелчком мыши обновлять через Интернет программу и набор сигнатур вирусов. Имеется Мастер по борьбе с вирусами. Он выдает информацию об обнаруженном вирусе и позволяет выбрать автоматическое либо ручное пошаговое удаление вируса. Антивирусные базы обновляются ежедневно. Недостаток программы — сложность настройки.

Довольно широкое распространение получила антивирусная программа *Avast*, которая бесплатно распространяется для использования на домашнем компьютере. Для установки Avast нужно скачать его последнюю версию, инсталлировать на компьютере, зарегистрироваться на сайте производителя и, получив лицензионный код, в течение 60 дней ввести его в программу. Возможности Avast:

- высокий уровень выявления вирусов, троянов и червей;
- резидентный и обычный сканер;
- сканирование архивов;
- проверка входящей и исходящей электронной почты;
- проверка файлов непосредственно из Проводника Windows (щелкнув по его ярлыку правой кнопкой мыши и выбрав команду **Сканировать...**);
- изолирован от операционной системы, что обеспечивает большую безопасность работы;
- понятный интерфейс, есть русский язык;
- большое количество настроек;
- автоматические или запланированные обновления вирусных баз. Возможность обновления вручную.

Широко применяется программа-доктор *Dr.Web*. Она входит в состав антивирусного пакета DSAV, который распространяет фирма «Диалог-Наука». Программа-ревизор ADINF проверяет диски, программа Dr.Web анализирует новые и измененные файлы и выдает на экран сообщения и запросы.

После запуска Dr.Web открывается оболочка с меню, расположенным в верхней строке экрана. Для проверки следует выбрать пункты меню **Тест** → **Тестирование** и указать путь для выполнения теста. После ввода откроется окно тестирования. С помощью пункта **Настройка** нужно задать общие установки тестирования, уровень эвристики, высоту экрана, цветовую схему и т. д., а также действия при обнаружении зараженных файлов. После проверки выдается окно отчета о выполнении тести-

рования. Если обнаружен вирус, Dr.Web выведет отчет красным цветом.

Основные правила лечения файлов и системных областей дисков:

- запрещается выполнять непродуманные действия, которые могут привести к заражению ПК;
- если вирус еще не активизирован, то следует немедленно выключить компьютер;
- при заражении ПК вирусом компьютер следует перезагрузить и начать его лечение с дискет;
- все операции по обнаружению вируса и лечению ПК должны выполняться с эталонной, защищенной от записи дискеты с ОС;
- при лечении поиск вирусов следует проводить во всех файлах (например, задавая в программе Dr.Web параметр `/AL`).

Для лечения файлов на всех жестких дисках следует использовать следующие команды:

```
drweb * /CL /RV /HI /AR /HA1 /CU /DL /AL.
```

Для лечения файлов на одном жестком диске следует вместо звездочки (*) указать имя диска (например, C:). На рис. 2.8 представлен логотип антивируса Dr.Web.

Рис. 2.8. Логотип программы Dr.Web

Пользуются популярностью антивирусные программы Лаборатории Касперского.

Antiviral Toolkit Pro — признан во всем мире как один из самых надежных антивирусов. Прост, но обладает большими возможностями для борьбы с вирусами: эвристический механизм, избыточное сканирование, сканирование архивов и упакованных файлов и др. Лаборатория Касперского внимательно следит за появлением новых вирусов и своевременно выпускает обновления антивирусных баз. Имеется резидентный монитор для контроля за исполняемыми файлами.

Антивирус Касперского специально разработан для защиты персонального компьютера. Простые интерфейсы, централизованное управление, автоматическая работа позволяют продукту функционировать в фактически фоновом режиме до тех пор, пока компьютеру и данным не грозит опасность. В случае попытки проникновения вируса продукт автоматически определит источник угрозы и эффективно нейтрализует ее при минимальном участии пользователя.

Антивирус Касперского имеет специальные средства защиты — он проверяет входящую и исходящую почту, контролирует базы почтовых сообщений, обеспечивает безопасность подключений к Интернету и любых устанавливаемых программ.

Антивирус Касперского является последним технологическим достижением Лаборатории Касперского в области защиты домашнего компьютера от вирусных угроз. Возможности: 100%-ная защита от макровирусов. Защита даже от неизвестных вирусов. Надежный контроль целостности данных. Комплексная проверка почтовой корреспонденции. Защита мест хранения данных. Проверка памяти запущенных программ.

Часто используют следующие антивирусы.

NOD 32 Antivirus System обеспечивает хорошо сбалансированную безупречную защиту персональных компьютеров и корпоративных систем, работающих на платформах MS Windows 95/98/ME/NT/2000/2003/XP, UNIX/Linux, Novell, MS DOS, а также для почтовых серверов MS Exchange Server, Lotus Doino и др. Вирусы, черви, троянские программы и другой вредоносный код находятся на безопасном расстоянии от данных. Передовые методы обнаружения, используемые в программном обеспечении, защищают даже от будущих потенциальных опасностей и от большинства новых червей и вирусов.

Главным преимуществом NOD 32 является его быстрая работа, низкое потребление системных ресурсов, способность ловить почти 100 % вирусов.

AntiVir Personal Edition для Windows 98/ME/NT/2000/XP определяет и способен удалить более 50 тыс. вирусов. Имеет встроенную систему автообновления через Интернет.

Naomi (ОС Windows 9x/ME/NT/2K/XP) — утилита контролирует содержимое, загружаемое из Интернета. Такая фильтрация осуществляется по ссылкам и ключевым словам (поддерживается 10 языков), отображаемым на веб-странице. Программа не нуждается в настройках, можно только задать пароль, чтобы было

невозможно отключить фильтрацию интернет-контента без ввода соответствующего пароля. Рекомендуется применять в учебных учреждениях.

McAfee VirusScan (производитель McAfee Associates) — один из наиболее известных антивирусных пакетов. Очень хорошо удаляет вирусы, но не так хорошо обнаруживает новые разновидности файловых вирусов. Легко и быстро устанавливается. Можно сканировать все файлы или только программные, распространять или не распространять процедуру сканирования на сжатые файлы. Имеет много функций для работы с сетью Интернет.

McAfee PreScan (ОС Windows 9x/ME/NT/2K/XP) — антивирус, который начинает работать во время загрузки MS Windows. Продукт имеет богатую антивирусную базу, поддерживает сканирование и очистку FAT и NTFS, а также сканирование сменных устройств. Антивирусные утилиты, которые сканируют систему во время загрузки ОС до загрузки остальных приложений, увеличивают шанс обнаружения вредоносного ПО до того, как оно успеет принести вред.

Panda Antivirus Platinum — это инновационное антивирусное решение, защищает компьютер и от вирусов, и от хакеров. Благодаря таким встроенным функциям, как межсетевой экран и блокировщик скриптов, программа гарантирует защиту от сетевых вирусов, хакеров и других опасностей, связанных с сетью Интернет.

Panda Platinum Internet Security (производитель Panda Software) комплексно обеспечивает информационную безопасность. Включает защиту от вирусов, хакеров, спама, программ-шпионов и других интернет-угроз. Ключевые характеристики:

- антивирус — обнаруживает и уничтожает все типы вирусов во время получения/отправки электронной почты, загрузки файлов или работы в Интернете. Благодаря усовершенствованному механизму UltraFast антивирус работает незаметно;
- брандмауэр — превращает компьютер в крепость, защищая его от хакеров. Защищает от программ-шпионов, блокируя их на веб-страницах.

Symantic Norton AntiSpam (производитель Symantec) — позволяет предотвратить попадание ненужных сообщений в почтовый ящик пользователя. Программа, совместимая с любым почтовым клиентом POP3, производит многоуровневую фильтрацию входящих сообщений электронной почты, выявляя и помечая спам;

при этом вся нужная корреспонденция доставляется без задержки. Кроме того, Norton AntiSpam отсекает рекламные заголовки и всплывающие окна при работе в Интернете.

Контрольные вопросы

1. Что такое программа-архиватор?
2. Какие программы-архиваторы вы знаете?
3. Перечислите режимы помещения файлов в архив ZIP.
4. Как создать архив файлов, используя программы для ОС Windows?
5. Как разархивировать файлы, используя программы для ОС Windows?
6. Как классифицируют компьютерные вирусы?
7. Какие антивирусные программы вы знаете?
8. Как вылечить зараженные диски, файлы, оперативную память?

Глава 3
АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

3.1. Алгоритмизация

Алгоритм — одно из фундаментальных понятий программирования, он включает совокупность правил, указаний, сформулированных на некотором языке и направленных на достижение некоторой цели (пример — рецепт, программа).

Слово «алгоритм» происходит от algorithmic — латинской формы написания имени великого математика IX в. аль Хорезми, который сформулировал правила выполнения основных арифметических действий над многозначными числами.

С понятием «алгоритм» тесно связано понятие *«исполнитель»*. Чтобы достичь цели, алгоритм должен быть кем-то или чем-то исполнен. Это может быть человек, механическое устройство, робот, компьютер и др., способные понимать и выполнять команды алгоритма. Каждый исполнитель имеет свою систему *команд* — конечный перечень доступных пониманию указаний на выполнение определенных действий. Эта совокупность команд называется *системой команд исполнителя* (*СКИ*).

Основное свойство алгоритма — формальный характер работы исполнителя. Исполнитель действует формально (не вникая в содержание, выполняет некоторые правила, инструкции), но получает требуемый результат. Следовательно, алгоритм формализует процесс решения задачи, позволяя механически использовать команды алгоритма в указанной последовательности.

Итак, *алгоритм* — понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение поставленной цели. На формальном характере работы основан принцип программного управления работой компьютера, поскольку *программа* — это и есть алгоритм, представленный на языке программирования.

Алгоритм должен быть *понятным*, *дискретным* (состоять из отдельных законченных команд), *результативным* (выполняться за конечное число шагов), *определенным* (каждая команда и правило алгоритма должны быть четкими, однозначными, не оставляющими места для произвола).

3.1.1. Виды алгоритмов и способы их записи

Существует два вида алгоритмов: работы с величинами (числовыми, символьными, логическими) и работы «в обстановке» (например, робот). Они должны быть записаны на алгоритмическом языке, т. е. представлены в виде графического и/или словесного описания, таблицы, последовательности формул, на учебном алгоритмическом языке, языке программирования. Иногда их записывают на псевдокодах (специальных языках). *Алгоритмический язык* — это система обозначений и правил для единообразной и точной записи алгоритмов и исполнения их. Язык записи алгоритма должен быть понятным для человека.

Текстовая форма описания алгоритма ближе к языкам программирования, но, в отличие от последних, строгого синтаксиса в алгоритмическом языке (АЯ) нет. Для структурирования текста алгоритма в АЯ используют строчные отступы. Все конструкции одного уровня вложенности записываются на одном вертикальном уровне. Вложенные конструкции смещаются относительно внешней конструкции вправо. Например, запись и принцип исполнения компьютером алгоритма задачи сложения двух чисел будет выглядеть так, как показано на рис. 3.1.

Наиболее распространен специальный графический язык описания алгоритмов — *структурные схемы*. *Схема* в стандарте определена как «графическое представление определения, анали-

```
Алг  Сложение
цел A, B, C
нач
   ввод A
   ввод B
   C:=A+B
   вывод C
кон
```

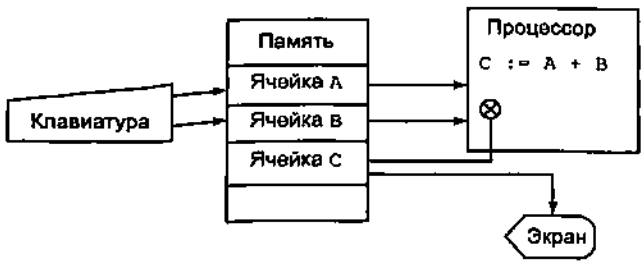


Рис. 3.1. Структурная схема исполнения компьютером вычислений алгоритма

за или метода решения задачи, в котором используются символы для отображения операций, данных, потока, оборудования и др.». Могут быть схемы данных, программ, работы системы, взаимодействия программ, ресурсов системы. Схемы обеспечивают наглядность, читаемость, отображение последовательности выполняемых процессов. Схема — это ориентированный граф, стрелками (или линиями) указывающий порядок исполнения команд алгоритма, а вершины (события) такого графа представлены геометрическими фигурами, которые называются символами. Например, (рис. 3.2): начало и конец записи алгоритма обозначаются овалом, данные (носитель которых не определен) — параллелограммом, процесс (обработка данных любого вида) — в виде прямоугольника, решение (или функция переключательного типа, например условие) — в виде ромба и т. д. Стандарт на этот специальный графический язык дан в ЕСПД (Единая система программной документации — ГОСТ 19.701—90).

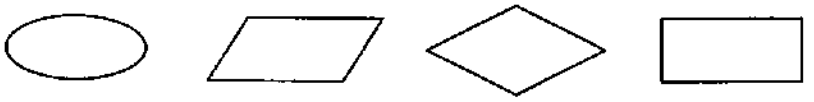


Рис. 3.2. Графические символы записи алгоритма в структурных схемах

В основе построения алгоритмических структур лежит теорема *структурного программирования*, включающая следующие основные принципы:

1) всякий реальный алгоритм может быть построен с использованием трех базовых элементов: следования, ветвления и цикла;

2) любая алгоритмическая структура, состоящая из базовых элементов, может быть представлена как единый процесс;

3) алгоритм проектируется по нисходящей схеме;

4) поэтапное уточнение или пошаговая детализация алгоритма.

В графической форме базовые вершины могут быть одного из трех типов:

функциональная (один вход и один выход); предикатная (один вход и два выхода); объединяющая (два выхода и один вход, передающий управление от первого из двух выходов). Из данных элементарных схем можно построить четыре схемы основных алгоритмических структур, имеющих особое значение для практики алгоритмизации (рис. 3.3).

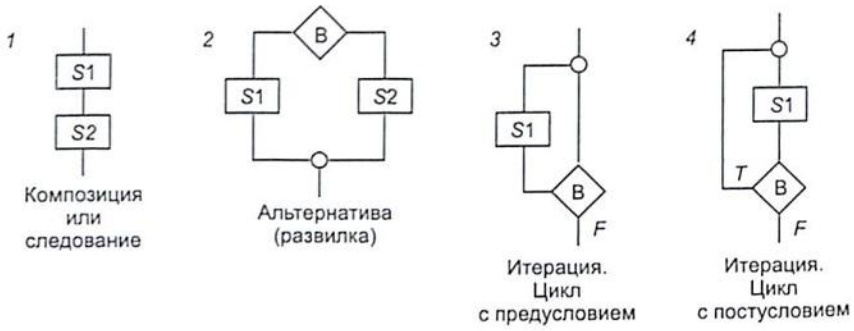


Рис. 3.3. Схемы основных алгоритмических структур

Первая схема на рис. 3.3 — самая простая структура алгоритма — последовательность, когда команды следуют одна за другой в естественном порядке. Такой алгоритм называется *линейным*. Здесь *S*1 и *S*2 — некоторые серии команд для использования (*S* — функциональные вершины).

Вторая схема — структура, в которой порядок следования команд определяется в зависимости от результатов проверки некоторых условий. Такой алгоритм называется *разветвляющимся*, здесь действия записываются не подряд, а в зависимости от выполнения (невыполнения) некоторого условия. *B* — условие (предикатная вершина), в зависимости от истинности (*T*) или ложности (*F*) которого управление передается по одной из двух ветвей. Объединяющая вершина обозначена символом *O*.

Третья и четвертая схемы — структуры, в которых получение результата обеспечивается *многократным повторением* одних и тех же операций. Это *циклические* алгоритмы с предусловием и постусловием.

Схему алгоритма можно нарисовать, например, вызвав через меню текстового процессора MS Word команды **Вид** → **Панели инструментов** → **Рисование**, с помощью *автофигур* (группа **Блок-схема**), где представлены начертания всех стандартных фигур и их обозначения. Надписи в фигурах следует создать с помощью команд **Вставка** → **Надпись**.

Типовая структурная схема алгоритма *линейной* структуры включает следующие последовательные действия: начало, ввод данных, действия, вывод результата, конец.

На рис. 3.4 представлена структурная схема линейного алгоритма для решения следующей задачи: «Найти энергию магнитного поля при заданных значениях магнитного потока, силы

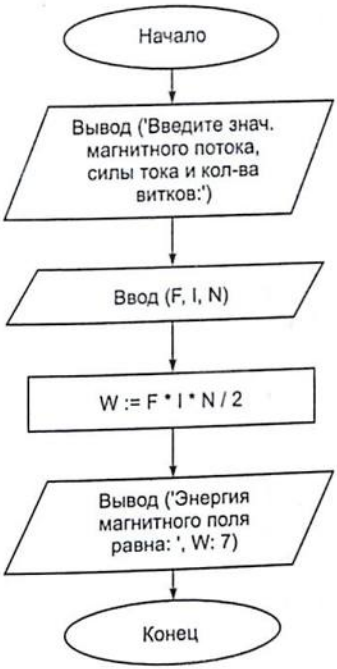


Рис. 3.4. Структурная схема алгоритма к задаче линейной структуры

тока и количества витков». (Программу к этой задаче см. в разделе 3.2.)

В схеме *разветвляющегося* алгоритма проверку условия выполняет логический блок, который изображается ромбом. Внутри него указывается проверяемое условие (отношение), и имеется два выхода: Да и Нет. Если условие истинно, то выходим на Да, если ложно — то на Нет (рис. 3.5).

На рис. 3.6 представлена схема циклического алгоритма.

Для составления алгоритма следует:

- осмыслить условия (требования) задачи, т. е. выяснить, каковы исходные данные, какие данные допускаются в задаче, что является результатом;
- составить строгую формулировку задачи по следующей форме: задача, аргументы, ограничения, результаты;
- записать описательную часть алгоритма и наметить план решения, т. е. процедурную часть, составив ее в произвольной словесной форме;
- переписать алгоритм, используя допустимые для предполагаемого исполнителя команды, оставляя неясные места;

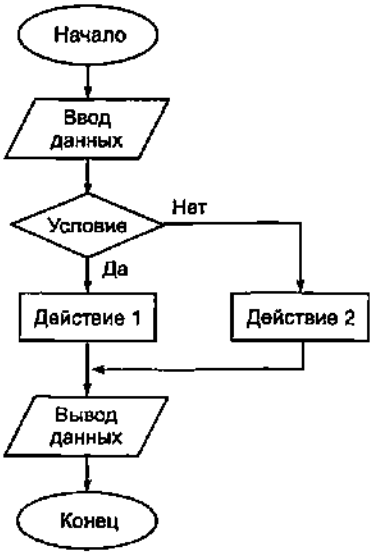


Рис. 3.5. Типовая структурная схема разветвляющегося алгоритма

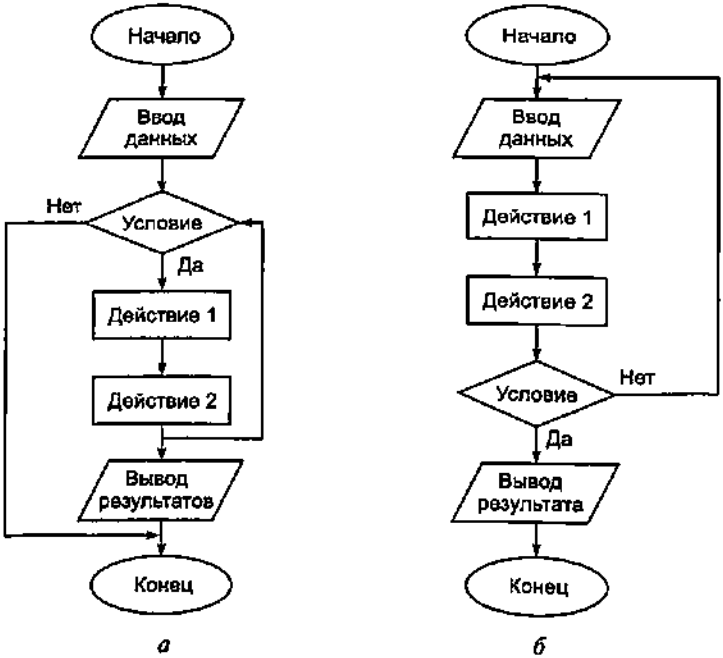


Рис. 3.6. Типовая структурная схема циклического алгоритма: *а* — цикл с предусловием; *б* — цикл с постусловием

- многократно переписывать алгоритм, каждый раз уделяя внимание вопросам, оставшимся от предыдущего шага, более мелким деталям, пока не получится текст, по которому можно писать программу для исполнителя (необязательно компьютера).

Важнейший прием алгоритмизации — декомпозиция задачи, т. е. выделение в исходной задаче некоторых более простых подзадач. Алгоритмы решения таких подзадач называются *вспомогательными алгоритмами*, а реализующие их программы — *подпрограммами* (процедурами). Решение исходной задачи разбивается на несколько алгоритмов: основной и вспомогательные. Как правило, в основном алгоритме происходит многократное обращение к вспомогательному.

Рекурсия — способ задания функции, при которой значение ее (при определенном значении аргументов) выражается через уже заданные значения функции (при других значениях аргументов), т. е. это метод вычисления функции, процедуры или решения задачи посредством той же функции, процедуры и т. д.

Рекурсивный алгоритм — алгоритм, в котором можно обратиться к уже заданным определенным образом значениям последовательности команд (к самому себе). Такие алгоритмы используют для возвращения назад, позволяют эффективно использовать метод проб и ошибок. Так, в случае отсутствия возможности осуществить очередное действие возвращаются назад и возобновляют поиск дальнейших действий. Этот процесс можно осуществить по рекурсивному алгоритму (например, следующему):

```
Алг Выбор
Нач
     инициализация начального действия
repeat (повторять до тех пор) выбор очередного действия
     if подходит then его запись;
     if решение не полное then RETR;
     if неудача then стирание оператора и возврат на
         предыдущий
until (пока) удача or (или) нет действия
кон
```

Такая рекурсивная процедура поможет правильно построить программу.

Рекурсивные алгоритмы, например, позволяют строить регулярные образы, постепенно образующие красивые узоры. Узор создается из серии определенным образом задаваемого мотива.

Например, кривые Серпинского первого и второго порядка можно разбить на части, которые соединяются четырьмя отрезками. Эти четыре части кривой представляют одну и ту же ломаную, поворачивающуюся каждый раз на 90°.

Контрольные вопросы

1. Что такое алгоритм?
2. Что такое система команд исполнителя?
3. Какие вы знаете виды алгоритмов?
4. Что такое алгоритмический язык?
5. Каков графический язык описания алгоритмов?
6. Какие вы знаете графические символы записи действий алгоритма?
7. Какая теорема лежит в основе построения алгоритмических структур?
8. Какие вы знаете структуры алгоритмов?
9. Дайте характеристику алгоритмическим структурам.
10. Что такое рекурсия и рекурсивный алгоритм?

3.2. Технология программирования

Технология программирования — это совокупность методов и средств для разработки программного обеспечения. В технологии должны быть определены последовательность выполнения операций, условия, при которых выполняется каждая операция, описание самих операций: исходные данные, нормативные документы, в том числе стандарты, критерии и методы оценки, результаты и др.

Любая задача начинается со словесного описания, которое называется условием задачи, а в программировании — спецификацией задачи. Далее условие следует формализовать, т. е. так записать спецификацию, чтобы можно было ее решить на компьютере. Формализацией условия начинается этап, который называется математической формулировкой задачи. Выводятся формулы и выбираются методы решения задачи, т. е. строится математическая модель задачи. Далее в виде алгоритма строится последовательность стандартных действий, выполнение которых даст искомый результат. На выбранном языке программирования проектируется и пишется программа для компьютера. Затем

выполняется ее тестирование и отладка на компьютере. Выполняются необходимые расчеты. Таким образом, решение задачи проходит следующие этапы (рис. 3.7).

Рис. 3.7. Схема решения задачи на компьютере

Технология программирования предусматривает *переход от алгоритма к программе*. Программы пишут специалисты — программисты, т. е. люди, подготовленные в области создания программ. Каждый тип процессора имеет свою систему команд, из которых и создаются компьютерные программы. Но машинные коды сложны для восприятия человеком, поэтому постепенно и стали развиваться языки, более приближенные к естественным. Программисты стали писать программы (*программировать*) в удобной форме, на понятных им языках программирования. А написанные специальные программы (трансляторы), которые могут прочитать то, что написал программист, и перевести в машинные коды, стали инструментом программирования.

Постепенно программы накапливались, особенно на процессы, выполняемые наиболее часто, например ввод данных с клавиатуры, вывод информации на экран и др. Такие небольшие программы (процедуры) называют стандартными. В настоящее время программист при создании большой программы обращается к библиотекам стандартных программ, извлекает необходимые программы-процедуры и включает их в свою программу. Это значительно облегчает процесс программирования.

Под *программированием* понимается совокупность процессов, связанных с разработкой программ и их реализацией.

В зависимости от назначения программирование подразделяют на:

- системное — разработка средств общего программного обеспечения, в том числе операционных систем, вспомогательных программ, пакетов программ общесистемного назначения, например автоматизированных систем управления, систем управления базами данных и т. д.;
- прикладное — разработка и отладка программ для конечных пользователей, например бухгалтерских, обработки текстов и т. п.

В зависимости от метода программирование подразделяют на:

- процедурное — метод, в соответствии с которым программы пишутся как перечни последовательно выполняемых команд. При этом используются процедурно-ориентированные языки программирования, например, PL/1, Алгол-68, Паскаль, Си;
- структурное, модульное — метод написания программ небольшими независимыми структурированными частями (модулями), каждый из которых связан с какой-либо процедурой или функцией. Результирующая программа организуется в виде совокупности взаимосвязанных по определенным правилам модулей. Это упрощает разработку сложных программных продуктов и их тестирование. Языки для модульного программирования Турбо Паскаль (Turbo Pascal), Си++, Ада, Модула (Modula);
- декларативное — метод, предназначенный для решения задач искусственного интеллекта. В указанном контексте программа описывает логическую структуру решения задачи, указывая преимущественно, что нужно сделать, не вдаваясь в детали. Используются языки программирования типа Пролог;
- параллельное — разработка программ, обеспечивающих одновременное выполнение операций, связанных с обработкой данных;
- объектно-ориентированное (ООП) — метод, основанный на использовании концепции объекта, абстрагирующего конкретные его реализации в предметной области. При этом данные тесно связываются с выполняемыми над объектами процедурами;
- функциональное — метод, основанный на разбиении алгоритма решения задачи на отдельные функциональные модули, а также описании их связей и характера взаимодействия. Для этого наиболее широко используются языки Норе и М, частично Си и другие языки;
- эвристическое — метод, основанный на моделировании мыслительной деятельности человека. Используется для решения задач, не имеющих строго формализованного алгоритма или связанных с неполнотой исходных данных;
- компонентное — метод создания ПО путем сбора объектов-компонентов (физически отдельно существующих частей ПО), взаимодействующих между собой через стандар-

тизованные двоичные интерфейсы, в библиотеки или исполняемые файлы.

Программисты разрабатывают все более мощные и сложные программы. Современные методы разработки качественных программ существенно сокращают время и затраты на их создание. Это структурное программирование, пошаговое нисходящее уточнение, разбиение на функции (модули), объектно-ориентированное и компонентное программирование. Широко используется мультипрограммирование, когда несколько задач или программ одновременно разделяют между собой ресурсы компьютера.

Технология программирования развивалась вместе с развитием ЭВМ и языков программирования. Она прошла этап «стихийного» программирования, когда программы состояли из машинных кодов или ассемблеров и обрабатываемых данных.

В 1949 г. сотрудник высшего технического училища Пенсильванского университета Дж. Мочли поручил своим программистам создать программу, которая позволила бы машине воспринимать алгебраические уравнения, записанные в традиционной форме. Созданная система кодирования под названием Short Code (Короткий код) позволяла записывать уравнения не в двоичном виде, а с помощью двухсимвольных комбинаций. Например, уравнение вида $A = B + C$ могло быть введено в форме последовательности `00 S0 03 S1 07 S2`, где переменные *A*, *B*, *C* обозначены `S0`, `S1`, `S2`, а операции равенства и сложения — `03` и `07` (комбинация `00` определяет номер строки программы). Программист (кодировщик) мог затем дать указание присвоить переменной `S0`, например, значение 5, а переменной `S1` — значение 3. После этого компьютер уже сам определял значение переменной `S2`.

Этап *структурного программирования* начался в 1960—1970-х годах. В его основе лежит представление задачи в виде иерархии подзадач простейшей структуры (линейной последовательности, альтернативы, многократного повторения — цикла), реализуемых в виде небольших подпрограмм и модулей.

Модульное программирование — выделение групп подпрограмм, в которых используются одни и те же глобальные данные в отдельно транслируемые модули (библиотеки подпрограмм). Модули между собой связаны через интерфейс. Структурный подход в сочетании с модульным программированием позволяет разрабатывать надежные программы, содержащие не более 100 тыс. операторов.

Для сложного программного обеспечения на этапе 1980—1990-х годов стало применяться объектно-ориентированное программирование (ООП), когда программа представлена в виде совокупности объектов, каждый из которых служит экземпляром определенного класса. Класс — совокупность объектов, характеризующихся общностью свойств и применяемых методов обработки. Классы образуют иерархию с наследованием свойств. Для обеспечения взаимодействия программных объектов используются сообщения. Механизмы наследования, полиформизма, композиции, наполнения позволяют строить сложные объекты из простых. ООП позволило создать огромное количество библиотек классов, ориентированных на разные задачи. Многократное использование этих библиотек позволяет повысить производительность разработки программ и уменьшить количество ошибок.

ООП явилось прологом к визуальному программированию. Были созданы системы визуального программирования: Visual C++ Builder, Visual Basic, Visual Fox Pro, Delphi и др.

В основе ООП лежат следующие понятия: объект, класс, инкапсуляция, наследование, полиморфизм. Объект рассматривается как некоторая структура, соответствующая объекту реального мира и его поведению. Задача, решаемая с помощью методики ООП, описывается в терминах объектов и операций над ними. Программа при таком подходе представляется в виде набора объектов и связей между ними.

Объекты — это динамически размещаемые структуры. Тип объекта называется классом. При создании нового типа объекта фактически создается класс. При создании новой программной конструкции на основе некоторого класса создается *экземпляр класса* — объект. Каждый объект содержит уникальную копию каждого поля, определенного в его базовом классе. Все объекты одного класса имеют одни и те же методы. Для создания и удаления объектов используются специальные методы, называемые *конструкторами* и *деструкторами*.

Переменная, обозначающая объект, фактически является указателем, ссылающимся на данные объекта в памяти. На один и тот же объект могут ссылаться несколько объектных переменных. Поскольку объектные переменные являются указателями, они могут содержать значение `nil`, указывающее, что объектная переменная не ссылается ни на какой объект. В отличие от указателей, объектная переменная для доступа к объекту не требует разыменования. Например, оператор `Edit1.Text := 'NewData';` при-

сваивает значение `'NewData'` свойству `Text` поля ввода `Edit1`. Оператор разыменования при этом не используется.

ОС Windows управляет событиями. Например, после щелчка по кнопке она генерирует событие, которое сопровождается рассылкой соответствующих сообщений. Компоненты и элементы управления, например в Delphi, фактически являются объектами, методы которых активизируются после получения сообщений от ОС.

Три фундаментальных составляющих ООП: инкапсуляция, наследование и полиморфизм.

Инкапсуляция означает объединение всех данных об объекте и характеристик его поведения в одном блоке. Объект содержит свойства и методы, при этом предоставляются средства для сокрытия данных.

Наследование позволяет расширять классы и способствует созданию родительско-дочерних отношений между объектами. Например, в приложении базы данных, в которой хранится информация о служащих компании, могут быть определены классы `Employee` (Служащий) и `Manager` (Менеджер). Класс `Employee` содержит информацию о служащих — фамилии, номера карточек социального страхования и др. Кроме того, для каждого менеджера-служащего нужна та же информация, что и для служащих, и необходимы некоторые дополнительные сведения. Следовательно, между этими классами существует логическое отношение: класс `Manager` образует подмножество класса `Employee`. Класс `Manager` (дочерний класс) наследует все свойства и методы класса `Employee`, однако, кроме наследованных, он имеет собственные свойства и методы.

Полиморфизм (дословно — способность проявляться во многих формах) означает, что программа может обрабатывать объект по-разному в зависимости от его класса.

Этап *компонентного* подхода и CASE-технологии начался с середины 1990-х годов. Компонентный подход лежит в основе технологий COM и CORBA. Для программиста элемент такой технологии обладает свойствами, методами и событиями, и его можно использовать для построения приложений. COM (Component Object Model) — компонентная модель объекта. Эта технология является развитием технологии связывания и внедрения объектов OLE, используемой для создания сложных и составных документов в приложениях, работающих под управлением ОС Windows. COM позволяет использовать функции одной части ПО другой частью.

На базе COM для разработки программного обеспечения были созданы компонентные технологии:

- OLE-automation — технология создания программируемых приложений (например, ее поддерживает MS Excel);
- ActiveX — технология создания ПО как на одном компьютере, так и в распределенной сети. Создана на базе OLE-automation. ActiveX применяется для создания ПО в Интернете и написания программ в среде Delphi, C++ Builder, Visual C++ и др.;
- MTS (Microsoft Transaction Server — сервер управления транзакциями) — технология для стабильной и безопасной работы распределенных приложений, работающих с большими объемами передаваемых данных;
- MIDAS (Multitier Distributed Application Server — сервер многозвенных распределенных приложений) — технология для организации доступа к данным разных компьютеров с учетом сбалансированности нагрузки сети.

Технология CORBA (Common Object Request Broker Architecture — общая архитектура с посредником обработки запросов объектов) используется для создания распределенных приложений. Может работать на всех основных аппаратных и программных платформах.

Для ускоренной разработки ПО применяют технологию RAD (Rapid Application Development). Она позволяет максимально быстро получить первые версии ПО. Эту технологию используют в основном для относительно небольших проектов, когда не требуется высокий уровень планирования и проектирования. Для крупных систем с большим количеством уникального кода эту технологию нельзя применять.

Языки программирования (ЯП) для создания ПО могут быть выбраны, исходя из конкретных условий. Это связано со знанием программистом определенного языка, с наличием у организации-разработчика лицензионной версии системы программирования, решаемых задач и т. д. Более подробно ЯП рассматривается далее. О линии ЯП Паскаль — Турбо Паскаль — Delphi рассказано в источнике [5].

С самого начала следует выработать хороший стиль программирования и оформления программ. При создании объектов следует давать им содержательные имена (идентификаторы), использовать ступенчатую форму записи, каждый оператор разме-

щать на отдельной строке, давать ясные комментарии, размещать текст программы только в рамках экрана и др.

Программа — упорядоченный список команд (инструкций, включающих операторы и параметры) на языке программирования. Такая программа называется исходным текстом, или *исходным кодом*. Для реализации программа должна быть откомпилирована, в результате чего образуется *объектный код*, записанный в машинных кодах. Для подключения к программе необходимых стандартных процедур и функций используют *редактор связей*, который выполняет эту работу, извлекая из библиотек стандартных подпрограмм необходимые элементы и вставляя их в объектный код. Полученная программа называется *исполняемым кодом* и является уже *рабочей программой*, которую можно запустить на исполнение.

Разделение программы на части — важный элемент программирования, так как сложные программы почти невозможно создать без их разделения на относительно независимые части: *основная* часть программы и *подпрограммы* — именованные группы программных действий, которые могут быть вызваны из других мест программы. Подпрограммам дают уникальные имена. Они должны быть один раз описаны и могут быть многократно использованы.

Имена переменных, описываемых непосредственно после заголовка и до тела подпрограммы, называются *локальными*. Они допустимы для использования только в той подпрограмме, где они описаны. *Имена переменных*, описанных перед всеми подпрограммами, называются *глобальными*. Они доступны для использования как в подпрограммах, так и в основной части программы. Принято соглашение: внутри подпрограммы действуют локальные переменные, а снаружи — глобальные. Глобальные имена следует применять очень осторожно.

Подпрограммы условно разделяют на подпрограммы-функции и подпрограммы-процедуры. Но это относится не ко всем языкам программирования.

Подпрограмма-функция обязательно возвращает результат, который затем используется в программе. Тип результата должен быть задан. Для обозначения подпрограммы используют служебное слово `function`.

Задание компьютеру действий (т. е. функции, предназначения) в месте, отличном от места их использования, называют описанием подпрограммы.

Общий вид описания подпрограммы-функции на языке Паскаль [5] следующий:

```
function имя_функции (список_аргументов): тип результата
описания внутренних переменных подпрограммы
begin
     тело подпрограммы (операторы, задающие действия
     с использованием  формальных аргументов)
end;
```

Подпрограммы по своему строению напоминают простую программу.

Подпрограммы-процедуры не возвращают ни одного значения или возвращают более одного значения. Для обозначения таких подпрограмм используют служебное слово `procedure`, а задание типа результата отсутствует.

Общий вид описания подпрограммы-процедуры на языке Паскаль следующий:

```
procedure имя_ процедуры (список_аргументов);
описание внутренних переменных программы
begin
     тело подпрограммы
end;
```

Имя процедуры не может использоваться внутри процедуры как псевдопеременная.

Для записи исходного текста программы прежде всего необходимы три простые команды: присваивание, ввод, вывод. Команда *присваивания* служит для изменения состояния объектов алгоритма и обозначается, к примеру, в языке Pascal [5] символом := (например `X:=1`, `A:=B`). Команды ввода-вывода необходимы для связи ЭВМ с внешним миром. Эти команды относятся к разряду основных и реализованы в любой машине. Основной единицей программы, выполняющей определенные действия над данными, является *оператор*. Операторы бывают управляющие и обрабатывающие.

В программах для ЭВМ используются описания объектов следующих типов: целые, вещественные, литерные (символьные) и др. С каждым типом связан свой способ представления объекта в памяти ЭВМ.

Основная *структура программы* на языке Паскаль имеет следующие разделы:

```
Заголовок. Описание данных. Начало программы. Операторы.
Конец программы.
```

Как правило, в любой программе компьютер должен выполнять ввод, обработку, хранение и вывод информации. Составление программы осуществляют набором констант, переменных, операторов и др., как правило, указывая их в определенной последовательности. Часто программа состоит из основной (главной) части и подпрограмм.

Изучение программирования начнем с помощью команд языка Паскаль в среде Турбо Паскаль.

Составление программы осуществляют описанием переменных, констант и др., а также набором последовательно записанных команд.

Пример 3.1. Вывести на экран текст «Я хочу учиться».

```
Program Student;
begin
     writeln ('Я хочу учиться');
end.
```

В этой программе имеется только главная часть, которая находится между служебными словами `begin` (начало) и `end` (конец). Предложения языка программирования, задающие действия компьютера, называются *операторами* (т. е. заданиями операции над информацией). В примере оператор `writeln` — оператор вывода.

Операторы вывода:

- `write(b:m)`, где `b` — имя выводимой переменной, `m` — число позиций, константа или выражение целого типа;
- `write(b:m;n)`, где `n` — дробная часть значения выводимой переменной.

По формальным правилам языка то, что надо вывести на экран, помещают в круглые скобки ( ), а сам текст заключается в апострофы. Если задана команда `writeln`, то экран настраивается на вывод в следующей строке, если `write` — то будет предложен вывод в этой же строке экрана. В конце программы ставится точка.

Правила строги, если их нарушить, то система программирования выдаст сообщение об ошибке.

Символ ; (точка с запятой) используется для разделения предложений языка (операторов).

В Паскале *оператор ввода* с клавиатуры — `read`, за которым в круглых скобках указываются переменные через запятую. Завершение ввода — нажатие на клавишу **Enter**.

Собственно вычисление в программе выполняет *оператор присваивания*. Общий вид записи оператора присваивания: `a:= b`, где `a` — имя переменной, `b` — выражение.

В фигурных скобках { } даются *комментарии*, которые предназначены для человека и не воспринимаются компьютером.

Имя переменной обязательно начинается с латинской буквы или символа подчеркивания, а далее могут быть цифры, буквы, символы подчеркивания, например `Student_1`.

Тип переменных определяет, какое значение может принимать эта переменная и какие операции над этой переменной возможны. Наиболее простые типы: целые числа (`integer`), числа с дробной частью — вещественный (`real`), буквы, символы — символьный (`char`).

Для записи математических формул на языке программирования используются следующие обозначения:

- * (звездочка) — знак умножения;
- ^ — знак возведения в степень;
- / — знак деления и дополнительное применение скобок `(x+y)/(x-y)`;
- стандартные математические функции требуют обязательного указания аргументов в скобках — `Sin(x)`;
- корень квадратный записывается через функцию `sqr`;
- переменные с индексами будут рассматриваться далее (см. информацию о массивах);
- используют некоторые функции, например `Round` — для округления дробного значения до ближайшего целого, `Trunc` — для отбрасывания дробной части (Truncate — усечь, обрезать) и др.

Рассмотрим примеры создания *программ разной структуры* в соответствии со следующими этапами:

- постановка задачи;
- определение метода решения;
- составление сценария работы с ЭВМ;
- разработка алгоритма;
- перевод алгоритма в программу;
- ввод и тестирование программы;
- анализ полученных результатов.

Программирование задач линейной структуры. В программе линейной структуры все действия записываются строго последовательно, одно за другим, в соответствии с разработанным линейным алгоритмом.

Пример 3.2.

1. *Постановка (условие) задачи.* См. раздел 3.1.1.

Найти энергию магнитного поля катушки, если известны количество витков, изменение силы тока и изменение магнитного потока.

2. *Метод решения.*

$W_{м.п} = I \cdot I_2/2$, $Ei = -N \cdot \Delta\Phi/\Delta t$, $Ei = Ec$, $Ec = -I \cdot \Delta I/\Delta t$, $N \cdot \Delta\Phi = I \cdot \Delta I$, $I = \Delta\Phi \cdot N/\Delta I$. Подставим выведенное значение I в формулу энергии магнитного поля и после сокращения получим: $W_{м.п} = \Delta\Phi \cdot \Delta I \cdot N/2$. По этой формуле и будем вычислять энергию магнитного поля.

Программирование следует начинать со сценария — определения типа переменных, их разрядности, порядка ввода, вывода результата.

3. *Сценарий*:

а) Ввести количество витков катушки (N), изменение силы тока (ΔI) и изменение магнитного потока ($\Delta\Phi$). По выведенной формуле $W_{м.п} = \Delta\ \Phi \cdot \Delta I \cdot N/2$ найти энергию магнитного поля катушки и вывести на экран.

$\Delta\Phi \cdot \Delta I \cdot N$ — вещественные числа, разрядность — 7 знаков после запятой (для удобства обозначить $\Delta\Phi$, ΔI в среде программирования Турбо Паскаль F и I соответственно). F, I, N, W — вещественные переменные.

б) Записать алгоритм решения задачи в виде схемы (см. рис. 3.4).

в) Результат решения задачи вывести на экран.

```
Program Energy;
var                    {служебное слово для описания
                            раздела переменных}
F, I, N,W: real;       {тип переменных – вещественные}
begin
     Writeln('Введите значения магнитного потока, силы
               тока и количество витков:');
     read F, I, N;     { ввод переменных}
     W:= F*I* N / 2    {вычисление энергии}
     Writeln ('Энергия магнитного поля равна:',W:7)
end.
```

4. *Результат выполнения программы.* Запускаем программу на выполнение с помощью системной команды RUN. Исправляем ошибки, т. е. выполняем отладку программы.

Вывод на экране (например): «Введите значения магнитного потока, силы тока и количество витков: 0.0023 7.5 120».

Ввод: Вещественные числа.

Ответ на экране (например): «Энергия магнитного поля равна: 1.0E+00».

Программирование задач разветвляющейся структуры.

Пример 3.3.

1. *Постановка задачи.*

Решить уравнение $y = |x|$.

2. *Метод решения*: $y = x$, если $x \geq 0$; $y = -x$, если $x < 0$.

3. *Сценарий*: для начала надо ввести x, затем поставить условие $x \geq 0$, в зависимости от которого присвоить y значение x или $-x$, затем распечатать y (x, y — целые).

4. *Алгоритм на учебном языке*:

```
АЛГ Uravnenie (x,y: целые)
Нач
Вывод ( 'Введите x:')
Ввод (x)
Если x>=0 то y:=x
     иначе y:=-x
Вывод ('Ответ:', y)
Кон
```

5. *Программа*:

```
Program Uravnenie;
Var
x, y: integer;
begin
     Writeln('Введите x:');
     read (x);
     If x>=0 then y:=x
          else y:= -x;
     Writeln ('Ответ:', y)
end.
```

7. *Результаты решения задачи*:

Введите x:	Введите x:	Введите x:
9	0	–6
Ответ: 9	Ответ: 0	Ответ: 6

Программирование задач циклической структуры на Turbo Pascal. Для представления циклов в программах имеется три типа стандартных операторов: цикл с параметром, цикл с предусловием и цикл с постусловием.

Цикл с параметром используется, если какие-либо действия нужно выполнить определенное число раз. Между началом и концом цикла заключаются действия, выполняющиеся нужное число раз. После служебного слова `for` (для) стоит параметр (счетчик) цикла с указанием его начального и конечного значений и шагом изменения.

Форма записи цикла с параметром следующая:

```
for параметр: = начало to конец do оператор
    (при возрастании параметра с шагом 1);
for параметр: = начало downto конец do оператор
    (при убывании параметра с шагом -1);
```

где `параметр` — числовая переменная; `начало` — начальное значение параметра; `конец` — конечное значение параметра; `шаг` — величина изменения значения счетчика.

Рассмотрим пример создания алгоритма и программы, имеющий цикл с параметром.

Пример 3.4. Вывести на экран 5 раз ваше имя.

Алгоритм на учебном языке:

```
АЛГ
НАЧ цел k 'k – параметр цикла
нц для k от 1 до 5
     вывод ('Имя')
кц
КОН
```

Программа:

```
Program Imja;
const a:='Александр';
var k:=integer;
begin
     for k:=1 to 5 do
     writeln ('Мое имя', a:9);
end.
```

Программирование циклических задач с предусловием (While ... do) и постусловием (Repeat ... Until) см. в [5].

Как было отмечено ранее, *разработка программы* проходит следующие этапы:

1) анализ требований, предъявляемых к ПО;

2) определение спецификации;

3) алгоритмизация (проектирование);

4) кодирование (перевод алгоритма на язык программирования);

5) тестирование и отладка;

6) эксплуатация и сопровождение.

Имеется пять классов *средств программирования* (*программ*): редактирующие, транслирующие, загрузчики, моделирующие, отладочные.

Редактирующие программы облегчают создание исходной программы. Они оперируют с исходной программой как с текстом, предоставляя различные возможности для изменений в тексте программы.

Транслирующие программы позволяют получить из исходной объектную программу, т. е. программу на машинном языке. Они реализуют программу, преобразуя запись на языке высокого уровня в последовательность машинных команд. Имеется два вида транслирующих программ: компиляторы и интерпретаторы.

В результате трансляции создается исполняемый файл (имеет расширение .exe).

Интерпретаторы — программы, которые анализируют каждую инструкцию (каждый шаг) транслируемой программы отдельно, после чего преобразуют ее в машинный код и выполняют его. Если какой-либо фрагмент исходного кода повторяется, то интерпретатор будет снова анализировать и преобразовывать этот фрагмент перед его исполнением.

Компиляторы — программы, преобразующие в ходе непрерывного процесса весь файл исходного текста в машинный код, после чего осуществляется его выполнение. Если какой-либо фрагмент компилируемой программы повторяется, то компиляция повторно не выполняется. Компиляторы работают значительно быстрее, чем интерпретаторы, но для начинающих программистов легче обучаться на интерпретаторах.

Загрузчики переносят объектную программу из внешней памяти в оперативную память ЭВМ.

Моделирующие программы — межмашинные программы, позволяющие проверить (отладить) объектную программу без ЭВМ.

Программы создают (пишут) в определенной *среде программирования* на ПЭВМ. Среда включает инструментальную программную оболочку, облегчающую составление и отладку программ. Чем совершеннее среда программирования, тем проще и быстрее создавать в ней новые программы. Например, интегрированная среда Турбо Паскаль состоит из двух частей: компилятора с языка программирования Паскаль и инструментальной программной оболочки.

Отладочные программы облегчают отладку объектной программы на ЭВМ.

Отладка программы выполняется после подготовки текста программы, когда ее можно попытаться исполнить, т. е. откомпилировать и связать (если необходимо) с библиотекой стандартных процедур и функций, загрузить в оперативную память и передать ей управление. Это называется прогоном программы и реализуется командой RUN или сочетанием клавишам **Ctrl+F9**. Если нет ошибок, то все действия выполняются последовательно. Компилятор может обнаружить ошибки в программе. Их следует исправить. Если компилятор обнаружит ошибку, то действия прекращаются, восстанавливается окно программы, в верхней части которого красным цветом выводится сообщение об ошибке, а курсор устанавливается на строку программы, где ошибка была обнаружена. Это позволяет быстро исправить ошибки в тексте программы, отладить ее. Синтаксические ошибки можно быстро исправить с помощью команд меню **Compile**. Если ошибка математическая, то следует выполнить пошаговую отладку с помощью клавиш **F4**, **F7**, **F8**. Около переменной устанавливается курсор (**Ctrl+F4**), и следует ввести имя переменной.

Тестирование (*верификация*) — проверка правильности работы программы. При тестировании программы используются специальные наборы параметров, для которых задача решается достаточно точно. Если полученный результат тестирования с этими параметрами удовлетворяет программиста и заказчика, то считается, что программа работает корректно.

Сопровождение программы предусматривает анализ работы программы по результатам производственных испытаний с наборами реальных данных. Поэтому после тестирования программы и передачи ее заказчику следует обучить его работе с програм-

мой, давать консультации и устранять ошибки, выявленные в процессе производственной эксплуатации программы.

Независимо от назначения и области применения программ должны быть пройдены следующие стадии их разработки.

Стадия 1. *Техническое задание*: обоснование необходимости разработки программы; научно-исследовательские работы, разработка и утверждение технического задания.

Стадия 2. *Эскизный проект*: разработка эскизного проекта, утверждение эскизного проекта (предварительная разработка структуры входных и выходных данных, уточнение методов решения задач, разработка общего описания алгоритма решения задачи и технико-экономического обоснования, разработка пояснительной записки).

Стадия 3. *Технический проект* (*ТП*): разработка ТП, утверждение ТП (уточняют структуру входных и выходных данных, разрабатывают алгоритм решения задачи, определяют форму представления входных и выходных данных, определяют семантику и синтаксис языка, разрабатывают структуру программы, окончательно определяют конфигурацию технических средств, подготавливают план мероприятий по разработке и внедрению программ, разрабатывают пояснительную записку, согласовывают и утверждают ТП).

Стадия 4. *Рабочий проект*: разработка программы и программной документации, испытания программы (создают программу и выполняют ее отладку, разрабатывают программные документы в соответствии с требованиями действующих стандартов, разрабатывают, согласовывают и утверждают программу и методику испытаний, проводят предварительные государственные, межведомственные, приемосдаточные и другие виды испытаний, корректируют программы и программную документацию по результатам испытаний).

Стадия 5. *Внедрение*: подготовка и передача программы. Выполняют подготовку и передачу программы и программной документации для сопровождения и/или изготовления, оформляют и утверждают акт о передаче программы на сопровождение и/или изготовление, передают программы в фонд алгоритмов и программ.

Допускается объединять, исключать этапы работ и/или их содержание. Программное обеспечение, как правило, разрабатывают в двух видах: для функционирования ЭВМ и прикладного характера, включая пакеты прикладных программ общего назна-

чения (стандартные, типовые) и специального и/или профессионального назначения.

Для эффективного использования ПО необходимо разрабатывать грамотную, понятную пользователям программную документацию: руководство программиста, руководство пользователя, руководство системного программиста, пояснительные записки и др. Правила составления документов даны в действующих стандартах.

Контрольные вопросы

1. Что такое технология программирования?
2. Опишите последовательность решения задачи на компьютере.
3. Что такое программа и как она реализуется на компьютере в рабочую программу?
4. Какие бывают имена переменных?
5. Что такое подпрограмма-функция и подпрограмма-процедура?
6. Из каких частей (разделов) состоит структура программы?
7. Какие вы знаете операторы?
8. Что такое тип переменных?
9. Что включает среда программирования?
10. Какие понятия лежат в основе объектно-ориентированного программирования?
11. Что такое компонентное программирование?
12. Каковы стадии разработки программ?

3.3. Языки программирования

Язык программирования — совокупность основных символов (алфавит) и правил составления из них смысловых конструкций. Он должен обеспечивать взаимодействие человека с машиной и быть удобен и понятен им обоим.

В соответствии с представлениями об уровне абстракции, на котором формируется задача и процессы обработки данных в реальном устройстве (процессоре), говорят об *уровне* языков программирования. Можно выделить языки низкого и высокого уровня.

К *языкам низкого уровня* относят:

- машинные языки — языки кодов ЭВМ;
- машинно-ориентированные языки: — ассемблеры, мнемокоды.

Языки реально функционирующих процессов называются *языками низкого уровня* (например, язык для функционирования процессора). Написание программ на языках низкого уровня требует более высокой квалификации программиста. Первые языки программирования представляли собой машинные коды, были эффективны и удобны для работы ЭВМ, но весьма трудны для восприятия человеком и требовали от него длительного изучения и специальных навыков использования. Программист должен был писать действительные адреса ячеек памяти, в которых размещены данные, участвующие в операциях, результаты, а также адреса команд.

Язык *ассемблера* (*Assembler*) — машинно-ориентированный язык, использующий символическое представление машинного языка. Операции машинного языка записываются в форме с применением мнемоник (англоязычных аббревиатур). Имеется возможность объединения нескольких машинных команд в макрокоманды. Но программы на ассемблере все-таки громоздки и их написание сложно. Хотя для задач, связанных с аппаратной частью ЭВМ, они эффективны. Поэтому для решения указанных задач вставки на ассемблере до сих пор используют в программах на языках более высокого уровня. Использование машинных языков сохраняется еще в АСУТП (АСУ технологических процессов).

Потребность в более понятных и удобных языках для написания программ различного назначения привела к созданию *языков высокого уровня*, в которых для выполнения сложных действий пишут один оператор. В языках высокого уровня инструкции программы часто представляют собой обычный текст на английском языке с применением знаков арифметических или логических операций.

К языкам высокого уровня относят:

- проблемно-ориентированные (имеют средства для организации структур данных и описания алгоритмов и ориентированы на решение задач определенного класса): Фортран, Алгол, Кобол, Ада и др.;
- универсальные: Алгол-68, PL/1, Паскаль, QBasic, Си++, Си шарп и др.;
- языки проектирования программ (системы программирования) — в настоящее время имеют самый высокий уровень абстракции. Они расширяются не как языки описания процесса обработки данных, а как средства описания задач:

Visual Basic, Delphi, MS Visual C++, Borland, C++ Builder и др.;

- языки гипертекстовой разметки, например HTML — набор кодов, который вводится в документ для обозначения, скажем, связей между его частями. Команды HTML обеспечивают соединение сайтов и главных страниц Всемирной паутины при помощи гиперссылок и указывают веб-браузеру (программе навигации) способ расположения массивов данных;
- языки описания сценариев — макросы, в которых объединены отдельные команды, управляющие средой в соответствии с их списком-программой (например, состоящие из именованных последовательностей совокупности указанных нажатий клавиш при работе с пакетом Microsoft Office);
- языки моделирования систем — например, GPSS (General Purpose Simulating System), который позволяет автоматизировать при моделировании процесс программирования моделей. Язык построен в предположении, что моделью сложной дискретной системы является описание ее элементов и логических правил их взаимодействия.

При решении типовых задач обработки данных в АИС применяются языки высокого уровня и системы программирования. Автоматическое программирование — методы перевода с входного языка на машинный язык для работы по подготовке и программированию задач на ЭВМ — находит все большее распространение. Используется мультипрограммирование, которое обеспечивает возможность использования для решения разных задач одних и тех же ресурсов ЭВМ, а также параллельной работы нескольких программ.

Проблемно-ориентированные языки предназначены для решения определенного класса задач и не зависят от конкретного типа ЭВМ. Они содержат перечень типовых операций, используемых при решении данного класса задач, и их условные наименования в терминах, привычных для специалистов в данной области. Например, Фортран и Алгол созданы для решения математических задач, Кобол — для решения экономических задач и др. Эти языки появились в конце 1950-х годов и положили начало эпохе развития языков программирования высокого уровня.

На сегодняшний день универсальными языками программирования, для которых характерны многоплатформенность, реализация всех основных структурных алгоритмических конструкций

(условия, циклы), большие накопленные библиотеки подпрограмм и классов, являются Паскаль, Си, Си++, Бейсик, Модула, Ада, Java и др. Объектное представление программы использовано в новых версиях универсальных языков программирования: Object Pascal, Си++, Java и др. Некоторые языки, являясь универсальными, в то же время используются как проблемно-ориентированные: Алгол-68 — для научных задач и моделирования, PL/I, использующий многие свойства Фортрана, Алгола, Кобола, Паскаля, — для обработки больших массивов данных, Simula — для имитационного моделирования сложных систем, Бейсик — многоцелевой, символический, обучающий, Лисп — для работы со списочными структурами.

Для перевода этих языков на внутренний язык ЭВМ необходимо иметь специальные переводящие программы — *трансляторы*. Хотя время написания и отладки программы сокращается, но уменьшается эффективность использования характеристик конкретной ЭВМ при решении транслируемой программы.

Кроме универсальных выделяют группы специализированных языков:

- баз данных (например, FoxPro, Oracle и др.);
- создания сетевых приложений (например, MySQL, SQL Server и др.);
- создания систем искусственного интеллекта (например, MYCIN и др.);
- пользовательские (профессиональные среды пользователя).

История развития языков программирования. Впервые с потребностью в написании программ для вычислительной машины Ч. Бэббиджа столкнулась математик Ада Лавлейс (рис. 3.8), которая создала первую программу — для вычисления чисел Бернулли (введя циклы, условные переходы и др.). Она считается первым программистом, в ее честь назван язык *Ада* (1979).

Рис. 3.8. Ада Лавлейс (1815—1852)

Для первых ЭВМ, как указывалось выше, был создан в 1949 г. под руководством Джона Мочли язык *Short Code* (*Короткий код*). Для первого в мире большого цифрового компьютера «Марк I» группа Мочли, куда входила и программист Грейс М. Хоппер, стали писать подпрограммы — многократно используемые последователь-

ности команд, построенные так, чтобы из них можно было формировать крупные блоки внутри программы.

В 1950-е годы появившаяся программа-компоновщик А-0, поставлявшаяся вместе с компьютером «Юнивак», использовалась для создания других программ. По условному коду компоновщик осуществлял выборку нужной подпрограммы из библиотеки подпрограмм. Извлекая несколько подпрограмм, он мог сформировать целую программу.

С развитием электронно-вычислительной техники в 1950-е годы появляются и языки программирования высокого уровня — Алгол, Фортран и Кобол.

В фирме IBM (США) группой инженеров под руководством Джона Бэкуса был разработан язык программирования *Фортран* (FORmula TRANslation — переводчик формул), предназначавшийся для решения математических и инженерных задач. Язык прост в освоении, а потери в эффективности по сравнению с ассемблерами — минимальны. Программирование на этом языке стало доступно многим.

Первая версия языка *Алгол* (ALGorithmic Language) принята на совещании в Цюрихе (Швейцария), в котором принимали участие ведущие специалисты из США, в их числе Джон Бэкус. Язык предназначался для записи алгоритмов вычислительных задач. Он стал первым языком с блочной структурой.

В 1959 г. для решения коммерческих задач группой производителей и пользователей компьютеров был создан язык *Кобол* (COBOL — Common Business Oriented Language) — универсальный язык, предназначенный для бизнеса. Язык хорошо структурирован. Кобол-программа состоит из четырех самостоятельных, следующих в строго определенном порядке разделов: идентификаций (задает название программы и содержит справочную информацию), оборудования (приводятся параметры ЭВМ), данных (описывает обрабатываемые данные), процедур (операторы).

В 1960 г. на конференции в Париже представители США, ФРГ, Великобритании, Франции, Дании, Голландии и Швейцарии утвердили улучшенную версию языка — Алгол-60, — получившую широкое распространение. На ее основе в 1960-е годы были разработаны такие языки, как Simula-67 — первый объектно-ориентированный язык, в котором впервые были определены понятия объекта, класса и наследования, Алгол-68 — с усиленными средствами ввода-вывода, но оказавшийся слишком громоздким и не получивщий широкого признания; в 1970-е годы —

алголоподобные языки Паскаль, Ада, Си; в 80-е — Си++; к концу 1990-х годов — Си шарп (C#).

В 1964 г. объединенной комиссией фирмы IBM и разработчиками проекта Share Fortran был предложен язык *PL/1* (Programming Language One), вобравший в себя черты Фортрана, Алгола-60 и Кобола. Язык совершенствовался до 1966 г., был пригоден для описания алгоритмов научно-технических, экономических, информационно-логических задач и стал универсальным языком широкого применения. Программа на PL/1 состоит из последовательности операторов двух групп: исполняемые (ввод-вывод данных, логические и расчетные операции, присваивание вычисленных значений некоторым переменным и др.) и неисполняемые (для описания и передачи транслятору информации о свойствах определенных объектов программы). Но написание текста программы на этом языке было достаточно неудобно, так как структура текста осталась еще с тех времен, когда программы вводились в ЭВМ с помощью перфокарт (на специальных бланках, имеющих 80 позиций в строке). Кроме того, большое количество средств и разнообразие операторов привело к тому, что PL/1 оказался довольно сложным в изучении.

В 1961—1965 гг. Джоном Маккарти в Массачусетском технологическом институте был создан язык функционального программирования (программа описывает вычисление некоторой функции), который ориентирован на решение задач нечисленного характера. *Лисп* (LISP — List Processing Language) — символьный язык для обработки списков. Понятие «список» понимается широко и включает представление алгебраических выражений, графов, элементов конечных групп, множеств, правил вывода и многое другое. Основные типы данных в языке называются «атом» и «точечная пара». Лисп создавался как язык для исследований по проблеме искусственного интеллекта. Основой этого языка служит математическая теория алгоритмов и *рекурсивных функций*, т. е. вычисление значения функции через значения этой же функции от других элементов. Присваивания и циклов в функциональном языке нет. Лисп — универсальный язык, так как любой алгоритм может быть описан с помощью некоторого набора рекурсивных функций, что позволяет моделировать на ЭВМ сложные алгоритмы обработки данных, в том числе алгоритмы моделирования интеллектуальной деятельности людей. В 1984 г. вышел Common Lisp, а затем система Common Lisp Object System (CLOS). В дальнейшем были разработаны

программные продукты, основанные на Лиспе: Common Lisp, Mac Lisp, Inter Lisp, Standard Lisp, Common Loops — система для обработки знаний и программирования, New Flavors — система посылки сообщений, используемая в коммерческих целях, и др.

В 1965 г. в Институте кибернетики Академии наук Украинской ССР под руководством академика В. М. Глушкова для ЭЦВМ «Мир» был разработан язык высокого уровня *Алмир* (алгоритмический язык для машины «Мир»), транслятор с которого был постоянно встроен в машину. Язык был разработан на базе Алгола, но использовал символику русского языка и стал одним из первых (возможно, первым) национальным языком программирования.

В 1966 г. в Институте прикладной математики АН СССР В. Турчиным был создан функциональный язык *РЕФАЛ* (алгоритмический язык рекурсивных функций). Построенный на алгоритмах Маркова, он удобен для обработки текстов. Язык активно использует мощные средства преобразования списков на основе концепции распознавания по образу.

В 1970-х годах появилось непроцедурное (декларативное) программирование — программирование на функциональных и логических языках, — начавшее быстро развиваться в 1980—1990-х годах в связи с разработкой в Японии проекта создания ЭВМ пятого поколения — поколения интеллектуальных машин.

Программа на логическом языке вообще не описывает действия. Она задает данные и соотношения между ними. После этого можно задавать вопросы. ЭВМ перебирает известные (заданные в программе) данные и находит ответ на вопрос. Порядок перебора не описывается в программе, а неявно задается самим языком.

Классическим языком *логического программирования* считается *Пролог* (Prolog — Programming in Logic) — язык для создания систем искусственного интеллекта, разработанный в 1971—1972 гг. сотрудником университета в Лумини (Франция) Аланом Колмари. Пролог (Пролог++) — символьно-логическая система программирования, вначале предназначенная для решения теорем, сейчас используемая для поиска решений, связанных с искусственным интеллектом. Язык позволяет в формальном виде описывать различные утверждения, правила рассуждений, заставляет ЭВМ рассуждать и давать ответы на заданные вопросы. Программа состоит из некоторого множества отношений, а ее выполнение сводится к выводу нового отношения на основе за-

данных. Понятия и принципы языка основаны на понятиях математической логики и аппарата автоматического доказательства теорем, созданного в ходе исследований по искусственному интеллекту. У языка есть потомки: Parlog (ориентирован на параллельные вычисления, выпущен в 1983 г.), Delta и др.

С течением времени модифицировался Фортран, появлялись его новые версии (1958 г. — Фортран II, 1961 г. — Фортран III, 1962 г. — Фортран IV, 1966 г. — Фортран-66, 1977 г. — Фортран-77, 1984—1988 гг. — Фортран-88). Особенно широко использовалась версия Фортран IV. В 1966 г. Американская организация стандартов (впоследствии преобразованная в Американский национальный институт стандартов — ANSI) издала первый стандарт программирования, известный как Фортран-66. В 1977 г. вышел новый стандарт — Фортран-77, расширивший возможности языка, особенно для обработки текстов и работы с файлами. В 1991 г. вышел стандарт Фортран-90, дополненный некоторыми расширениями. Последним стандартом языка был Фортран-95. Он и в настоящее время применяется для решения инженерных и научно-технических задач. Появился даже Visual Fortran.

В 1965 г. на основе Фортрана профессора Дартмутского колледжа (США) Джон Кемени и Томас Куртц для обучения студентов разработали язык *Бейсик* (BASIC — Beginner's All-Purpose Symbolic Instruction Code — универсальный символьный программный код для начинающих). Язык прост и в то же время имеет широкие возможности для обработки символьной и графической информации. Характерные черты языка: диалоговый режим работы, нумерация строк, вещественный и символьный типы данных, управляющие конструкции, все переменные являются глобальными, наличие массивов. Язык нашел широкое применение и имеет много версий.

Одну из них в 1975 г. написали Билл Гейтс и Пол Аллен — основатели корпорации Microsoft. В 1980-х годах эта корпорация выпустила *QuickBasic*, обладающий большим быстродействием, простотой, компактностью. Но основная часть языка во всех версиях одна. Вот лишь несколько версий языка: Basic, GWBasic, Power Basic, Turbo Basic, QuickBasic (компилирующая система языка входит в состав MS-DOS, начиная с версии DOS 5.0), Visual Basic (для программирования в среде Windows).

В 1970 г. швейцарский ученый Никлаус Вирт разработал язык Паскаль для новой операционной системы UNIX, установ-

ленной на компьютере CDC 6000. Язык нашел широкое применение и послужил базой для создания паскалеподобных языков: Ада, Модула-2, Оберон — два последних были разработаны также Никлаусом Виртом. В 1975 г. Вирт и Йенсен выпустили классическое описание языка «Pascal User and Report». В 1983 г. появляется Турбо Паскаль фирмы Borland. Выпустив семь версий этого пакета, фирма Borland/Inprise стала выпускать систему визуальной разработки Delphi для ОС Windows.

Язык *Си* создан Д. Ритчи и Б. Керниганом в 1972 г. В 1979 г. Бьерном Страуструпом разработан «Си с классами», а в 1983 г. — Си++. Этот язык в основном является языком Си со специальными синтаксическими расширениями для определения и управления объектами, введена поддержка абстракции данных, проверка типов аргументов функций и макроподстановка функций. Язык, повсеместно эффективно используемый для объектно-ориентированного программирования, постоянно совершенствуется. Были разработаны языки, основанные на Си, такие как Objective-C, C-talk, Complete C.

В настоящее время существует несколько тысяч языков и систем программирования. Однако широкое распространение получили лишь несколько десятков из них.

Языки Си и Паскаль стали самыми популярными языками для *структурного программирования*, подняв его на новый уровень упорядоченности и производительности.

А с появлением их усовершенствованных версий начинается эра объектно-ориентированного программирования.

Язык *Ада* разработан группой французских ученых под руководством Ж. Ишбиа в период 1975—1980 гг. по заказу Министерства обороны США для создания бортовых систем управления военными объектами. Конечная версия международного стандарта языка была опубликована в 1987 г. Последний стандарт описывает версию языка Ада-95. Она является первой в мире объектно-ориентированной системой программирования, на которую был введен международный стандарт —ISO/les 8652:1985 (E). По своей структуре язык Ада очень похож на Паскаль и имеет возможность использовать неоднородные структуры, разделение памяти, реализовывать неявные функции преобразования типов. Язык ориентирован на тщательный контроль программ, чтобы надежность военных систем была максимальной. Ада располагает большим набором данных (например, ch — символьный, такого типа нет ни в Фортране, ни в Алголе-60), возможно-

стью строить новые типы данных, не предусмотренные ранее. Язык оказался достаточно универсальным и применяется не только для решения военных задач.

Язык *Модула-2* создан в 1979 г. Никлаусом Виртом. В языке сочетается простота Паскаля и средства модулизации. Главное свойство языка — хорошо продуманная структура программ. Характерной чертой является раздельная компиляция, позволяющая разрабатывать и хранить в библиотеках программы, которые можно использовать повторно. Язык прост для изучения и эксплуатации.

Язык *Оберон* создан Никлаусом Виртом в 1987 г. как попытка достичь идеала универсального языка программирования. В нем имеются средства обогащения комбинированных типов данных, средства объектно-ориентированного программирования — расширяемые записи. Язык обеспечивает строгий контроль на этапе трансляции. В 1992 г. Х. Мессенбек предложил расширенную версию языка — Оберон-2. В настоящее время семейство оригинальных Оберон-систем известно под названием Oberon V4.

В системах программирования (Delphi, C++ Builder фирмы Borland, Visual Basic, Visual C++ фирмы Microsoft и др.) и в средах программирования (Турбо Паскаль и др.) хорошо представлен визуальный интерфейс, позволяющий пользователям легко общаться с системой. Визуальная среда программирования включает средства разработки программ: компилятор, текстовый редактор, компоновщик, отладчик, справочную систему и библиотеку программ.

Языки проектирования программ — *системы визуального программирования* — имеют самый высокий уровень абстракции. Они являются самым современным и эффективным средством создания программных продуктов и рассматриваются как средства описания задач. В число наиболее популярных входят следующие системы программирования: *Visual Basic*, *Delphi*, *C++Builder*, *Visual C++*, базирующиеся соответственно на языках Бейсик, Паскаль и Си++. Как правило, они имеют интегрированную среду разработки (IDE — Integrated Development Environment), включающую текстовый редактор, конструктор форм и набор других инструментов, позволяющих значительно сократить объем работ по созданию программных продуктов.

В 1991—1993 гг. Microsoft создает первую визуальную среду программирования — *систему программирования Visual Basic*, по-

зволяющую достаточно просто создавать Windows-программы. Visual Basic обладает средствами визуального проектирования, упрощает программирование на языке Бейсик и использует все графические функции Windows.

Visual Basic стал одним из первых языков, поддерживающих событийно-управляемое программирование (Event-Driven Programming). Оно позволяет программисту не описывать каждый шаг, а только указать, как реагировать на различные события (выбор команды, движение или щелчок мыши и др.), создавая таким образом набор управляемых взаимодействующих процедур. Visual Basic связан с приложением Windows — Microsoft Office, он снабжен встроенными средствами поддержки баз данных Access. В Visual Basic появились инструменты, позволяющие упростить работу с Интернетом, быстро подключить указанные программы к Глобальной сети, придавая им функции навигаторов. В последнее время некоторые серверы и браузеры поддерживают язык Visual Basic Script.

Отдельно следует выделить язык *dBase* фирмы Borland, который представляет собой гибрид объектно-ориентированного и традиционных языков и применяется для реализации баз данных. При их создании реализуют объектный дизайн и обычные приемы обработки записей. Был создан целый ряд эффективных dBase-подобных систем управления базами данных — СУБД (FoxBASE, Paradox и др.).

В 1995 г. фирма Borland создает систему программирования Delphi, а фирма Microsoft — Visual C++.

Высший на сегодняшний день уровень языков — языки описания сценариев. При их использовании повышается производительность труда. Это уже система программирования, позволяющая общаться человеку с машиной напрямую с минимальной его подготовкой. Обеспечивается легкий доступ к множеству существующих объектов, есть возможность манипулирования тысячами объектов, облегчается труд программиста.

Языки описания сценариев — подвид пакетных файлов, подобных макросам, в которых объединены отдельные команды, управляющие операционной средой в соответствии со списком, являющимся программой. Основное назначение этих языков — связывать различные компоненты и приложения друг с другом. Здесь нашел применение бестиповой подход к описанию данных, что значительно упрощает установление связей между компонентами и ускоряет разработку прикладных программ. Эти языки предна-

значены для комбинирования компонентов, набор которых создается заранее с помощью графического интерфейса пользователя, Интернета или инфраструктуры разработки и технологии сценариев. Например, языки Perl, TCL, ActiveX, Java и др.

Perl (Practical Extraction and Report Language — язык для практического извлечения данных и составления отчетов) — язык для обработки произвольных больших текстов и файлов. Его часто используют для написания системных программ. С помощью этого языка можно не только сканировать текстовые файлы, но и обрабатывать двоичные данные. Синтаксис языка близок к синтаксису Си. Perl позволяет использовать регулярные выражения, создавать объекты, вставлять фрагменты программ на Perl в программы на Си и Си++, осуществлять доступ к базам данных. С помощью языка Perl можно взаимодействовать с веб-серверами, получать данные из формы HTML (см. далее) и др.

Популярность объектно-ориентированного языка Си++ привела к рождению множества новых языков для современного Интернета. С развитием Глобальной сети появилась проблема переноса программ с одной платформы на другую. Для ее решения и создания мобильных сетевых программных продуктов фирмой Sun Microsystems в 1991 г. был создан новый язык *Java*. Язык разработали Д. Гослинг, П. Нотон, К. Ворт, Э. Фрэнк, М. Шеридан, и первоначально назвали его OAK.

Java — универсальный, объектно-ориентированный язык, позволяющий описывать любые задачи, включая обслуживание веб-страниц. Для функционирования он требует компиляции и вспомогательных файлов. По синтаксису и технологии программирования очень похож на Си++. Программы, написанные на языке Java, независимы от архитектуры ЭВМ. Это достигается путем транслирования программ в промежуточный язык, называемый *байт-кодом*. Java имеет защиту от компьютерных вирусов и несанкционированного доступа. Главным отличием Java-программ, которые называются Java-апплетами, является использование библиотеки Java-классов, обеспечивающих разработку безопасных распределенных систем. В 2003 г. компания Sun объявила о появлении нового продукта Joe, предназначенного для существенного облегчения встраивания Java-клиентов в информационные системы Интернета.

Фирма Netscape для целей мобильного программирования создала язык LiveScript, позволяющий простые программы вклю-

чать в текст документов, написанных на HTML. Затем эта фирма с разрешения фирмы Sun переименовала LiveScript в *JavaScript*. С его помощью удобно создавать сценарии, связанные с определенными событиями. Но JavaScript работает только в составе страниц HTML.

HTML (Hyper-Text Markup Language) — языке гипертекстовой разметки, язык для программирования во Всемирной паутине, язык для написания программного кода для веб-страниц. Первую версию HTML в 1990 г. разработал Тим Бернерс-Ли — сотрудник Европейской лаборатории физики элементарных частиц. По существу, это язык форматирования, определяющий, как на веб-странице будут размещены текст, графика, таблицы. Можно также создавать ссылки между различными частями информации. Основу HTML представляет обычный текст, в который вставляются управляющие символы (теги). Получаем веб-страницу. Этот объект может содержать информацию различного вида, просмотреть которую можно с помощью специальных программ-браузеров (Internet Explorer, Netscape Navigator и др.). Найти веб-страницу можно с помощью универсального указателя ресурсов (URL — Universal Resource Locator). Для доступа к веб-странице используется протокол передачи гипертекста HTTP (Hyper-Text Transfer Protocol), т. е. сначала указывается протокол, а затем имя сервера и путь к файлу веб-страницы.

SQL — язык структурированных запросов, непроцедурный язык высокого уровня.

Позволяет решать задачи получения информации из таблиц БД и манипулировать данными в таблицах. Используется для создания интерфейса БД. Реализован в популярных СУБД. Стал стандартным языком запросов. В 1992 г. объявлен международным стандартом. SQL включает язык определения схемы (SQL-DDL) и язык манипулирования данными (SQL-DML). Язык позволяет задать только то, что нужно делать, а само исполнение отдельных операций возложено на СУБД. Особенность языка заключается в так называемой трехзначной логике. Кроме булевых значений «истина» и «ложь», выражение может принимать значение NULL (пустое значение) — специальный код, который используется при отсутствии данных. В операциях сравнения NULL используется, когда результат неизвестен. Может применяться как SQL:

- интерактивный или автономный (для непосредственного извлечения информации из базы или записи в нее);

- статический (для записи фиксированного исполняемого кода SQL);
- динамический (для генерирования кода SQL во время исполнения приложения, построения запросов в диалоговых средах, в графических средствах разработки приложений БД).

В конце 1990-х годов в фирме Microsoft под руководством Андерса Хейльсберга был разработан язык *Си шарп* (C#). Альфа версия выпущена в 2000 г., а в 2001 г. окончательная версия вошла в пакет Visual Studio.Net 7. Язык унаследовал лучшие свойства языков Си и Си++. Си шарп работает в среде исполнения .NET Framework, что обеспечивает переносимость программ в многоязыковое программирование. Программы на Си шарп могут быть скомпилированы в среде Visual C++.

Таким образом, хронологию развития наиболее распространенных языков программирования можно представить в такой последовательности (табл. 3.1).

В настоящее время программирование на персональных ЭВМ ведется в основном на языках Бейсик, Паскаль и Си (как правило, их последних версиях), а на больших и мини-ЭВМ все еще применяются языки Фортран и PL/1. Для перспективных ЭВМ нового поколения прототипом будущего языка программирования считается Пролог. Появление более быстрых ЭВМ и более совершенных языков описания сценариев, удобных графических интерфейсов пользователя и компонентных архитектур, популярность Интернета расширяют сферу применения языков описания сценариев для различных приложений.

Чтобы алгоритм стал понятен ЭВМ, его следует *закодировать* — перевести на строго формализованный язык, т. е. язык программирования. Кодирование — процесс достаточно механический, но требует твердого знания команд и синтаксиса языка программирования.

Алгоритм, записанный на языке программирования, называется *программой для ЭВМ*. Языки эти формальные, специально созданные для общения человека с компьютером. Каждый язык программирования имеет алфавит, словарный запас, грамматику, синтаксис и семантику. *Алфавит* — фиксированный набор символов, допускаемых для составления текста программы на этом языке. *Синтаксис* — система правил, определяющих допустимые конструкции языка программирования из букв алфавита. *Семантика* — система правил однозначного толкования отдельных языковых конструкций, позволяющих воспроизвести вы-

Таблица 3.1. **Хронология развития языков программирования**

Язык/год				
Фортран/ 1954—1957	***Алгол***/1958		***Кобол***/1959	
	Алгол-60/1960		***Кобол***, ***Фортран***, ***Алгол-60***	
Бейсик/1965		Simula-67	PL1/1963—1966	***Лисп***/1962
		Алгол-68		***РЕФАЛ***/1966
	Паскаль/»1970			***Пролог***/1971 ***Пролог++***/ 1972
		Ада/ 1975—1980		
		Модула-2/ 1979		
QuickBasic/ 1980-е		*Оберон*/ 1987—1988		*Common Lisp*/ 1985
		Си/1972		
	Турбо Паскаль/ 1983	*Си++*/ 1980—1983		
	Delphi/1995	*Java*/1991		**HTML**/1990
Visual Basic/ 1992		*JavaScript*/ 1992		**SQL**
		Си шарп/2000		

полнение процесса обработки данных. Если программа написана в машинных кодах, то она может сразу исполняться ЭВМ. Но писать такие программы сложно, они очень громоздки и плохо воспринимаются человеком.

Если формальный язык, на котором написана программа, не может сразу быть воспринят компьютером, то создают специальные программы (трансляторы: компиляторы или интерпретато-

ры), осуществляющие автоматический перевод программ с языка программирования в машинные коды.

Программы-компиляторы (разновидность трансляторов) выполняют основные этапы компиляции:

- вводят исходный код (специальные символы, слова и математические выражения);
- преобразуют его в объектный код;
- подключают с помощью редактора связей необходимые стандартные программные модули из библиотек;
- заставляют компьютер выполнять соответствующие команды на машинном языке (исполняемый код — рабочая программа).

Предположим, что исполнителем алгоритма будет ЭВМ. Компьютер, к сожалению, не воспринимает команды на естественном языке. Для управления компьютером необходимо освоить специальный командный язык. Внутри себя компьютер составляющими его микроэлектронными устройствами командует на внутреннем машинном языке, удобном для микроэлектроники.

Программы, записанные на любом языке программирования, сначала с помощью трансляторов переводят в *машинный код*. Его можно посмотреть во время работы с помощью программ-отладчиков, позволяющих находить ошибки в программе. Машинный код записывается в шестнадцатеричной системе счисления, двузначными числами или обозначениями (например, `33`, `41`, `45` , `C0`, `F6` и др.), каждому из которых отведен байт, находящийся в своей ячейке памяти. Таким образом, программа в машинном коде — это набор байтов, которые процессор понимает и различает: команды, числа, символы, адреса.

Более понятным программистам, чем машинный код, является специальный код — *код ассемблера*. Он записывается в виде мнемоник. Каждая команда — мнемоника (сокращенные слова английского языка). Например, машинный код, представленный числом 93, в виде мнемоники записывается как `EXCHG BX,AX` и означает — обменять: (`EXCHG`) содержимое регистров `BX` и `AX`. Иногда системные программы пишут на ассемблере, а потом переводят в машинный код — ассемблирование, или наоборот, чтобы легче было читать программу, выполняют дисассемблирование.

До 1980-х годов компьютеры использовались для математических и технических расчетов и действий с экономической информацией. Для этих задач разработаны проблемно-ориентиро-

ванные языки программирования. Дальнейшее развитие языков делало их более универсальными, позволяющими составлять программы для любых информационных действий, управления печатью, рисования чертежей, воспроизведения звуков и т. д. Сейчас квалифицированные пользователи используют два-три командных языка: для профессионалов — Си, для обучения — Паскаль, Бейсик.

Например, записи программ, написанных на трех языках (Бейсик, Паскаль и Си++) для решения одной и той же задачи — вычисление корней квадратного уравнения, — будут выглядеть так (листинги 3.1—3.3). Строки в программе на языке Бейсик нумеруются по возрастанию (как правило, числами, кратными 5 или 10), что удобно при использовании оператора безусловного перехода (`goto`). Язык имеет небольшой набор типов данных: целые (от –32 768 до +32 767), строковые (до 256 символов), с плавающей точкой обычной точности (7 цифр), с плавающей точкой двойной точности (16 цифр). В Бейсике есть средства обработки многомерных массивов, логические операции, управление форматом выводимых данных, обработка графики и др.

Листинг 3.1. Запись программы на языке Бейсик

```
10 PRINT "Вычисление корней квадратного уравнения"
20 PRINT "Введите коэффициенты уравнения"
INPUT a
INPUT b
50 INPUT c
60 d = b*b - 4*a*c;
70 IF d<0 GOTO 120
80 PRINT "Корни уравнения"
90 x1= (-b + sqrt(d))/(2*a) : x2: = (-b -sqrt(d))/(2*a)
100 PRINT x1, x2
110 GOTO 130
120 PRINT "Уравнение не имеет корней"
130 END
```

Листинг 3.2. Запись программы на языке Паскаль

```
Program KUrPas;                    {Заголовок программы}
Var                                {Описание переменных}
a,b,c: real;                       {Коэффициенты уравнения}
d,x1,x2: real;                     {Вспомогательные переменные}
```

```
begin {Начало программы}
     write ('Вычисление корней квадратного уравнения');
     writeln('Введите коэффициенты уравнения');
     read (a);
     read (b);
     read (c);
     d: = b*b - 4*a*c;       {Дискриминант}
     if d<0 then write(`Корней нет`) {если d<0, то вывод
на экран: Корней нет}
     else {иначе}
begin {Начало серии команд}
x1: = (-b + sqrt(d))/(2*a); {Вычисление корней}
x2: = (-b -sqrt(d))/(2*a);
write ('Корни квадратного уравнения x1 =' , x1; 'x2 =',
x2)
end;                          {Конец серии}
end.                          {Конец программы}
```

Листинг 3.3. Запись программы на языке Си++

```
# include<lost ream.h> /* сообщение компилятору, что он
                          должен включить
# include<math.h>      /* стандартные программы из
                          библиотек
# include<conio.h>
main ( )           /* функция без аргументов, ее
                      инструкции заключаются далее в { }
{
     int a, b, c, d, x1, x2;
     cout <<"Вычисление корней квадратного уравнения";
     cout <<"Введите коэффициенты уравнения";
     cin >>a;
     cin >>b;
     cin >>c;
     d: = b*b - 4*a*c;
     if (d<0) goto MET1;
     cout <<"Корни уравнения";
     x1: = (-b + sqrt(d))/(2*a);
     x2: = (-b -sqrt(d))/(2*a);
     cout <<"x1<<x2;
     getch ( );
     goto MET2;
     MET1:
     cout <<"Уравнение не имеет корней";
     getch ( );
     MET2:
}
```

Средства описания языков программирования. Как уже указывалось, язык программирования должен иметь алфавит, синтаксис и семантику. Правила, определяющие структуру текста, составляют синтаксис (грамматику). Правила, позволяющие извлекать из текста смысл, называются семантикой языка.

Для описания синтаксиса языка используется так называемый *метаязык* (язык для описания языка). В качестве метаязыка используют либо металингвистические формулы Бэкуса—Наура, либо синтаксические диаграммы. Джон Бэкус разработал специальную систему определений для языков программирования. Например, определяя элемент «цифра», он указывал:

<цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 .

Этот способ записи назвали нормальной формой Бэкуса, или БНФ. Датский астроном П. Наур внес некоторые уточнения, и форму стали называть формой Бэкуса—Наура, но сокращенное название БНФ осталось.

В БНФ каждое понятие определяется с помощью одной металингвистической формулы, в левой части которой указываются определяемые понятия, а в правой — описываются все допустимые структуры языка, соответствующие этому понятию. Левая и правая части формулы связываются знаком ::=, который читается «по определению есть». При перечислении дополнительных вариантов структур в правой части используется знак |, имеющий смысл «или». Все определяемые понятия языка программирования заключаются в угловые скобки < >, а неопределяемые исходные понятия языка называются *лексемами*. Лексемы могут использоваться в правой части формулы.

Например, определить понятие <целое число>, если алфавит языка программирования включает арабские цифры и знаки арифметических операций. Сначала определим понятие <цифра> (см. ранее). Затем введем понятие <целое без знака>, описав его следующей формулой:

<целое без знака> ::= <цифра> | <целое без знака> <цифра>

Это равносильно определению «любая последовательность цифр». Формула рекурсивна, так как <целое без знака> уже встречается в правой части, т. е. имеется ссылка на само себя. Теперь выведем формулу:

<целое число> ::= <целое без знака> | -<целое без знака> |
| +<целое без знака>.

Полученная цепочка из трех формул определяет понятие <целое число> через лексемы языка 0, 1, ..., 9, +, –.

С помощью металингвистических формул могут быть описаны все понятия языка программирования. Синтаксическая правильность текста проверяется в процессе его *синтаксического анализа* на основе металингвистических формул.

Способ описания синтаксиса на основе диаграмм основан на тех же принципах. На рис. 3.9 приводится изображение понятия «цифра» в виде диаграммы.

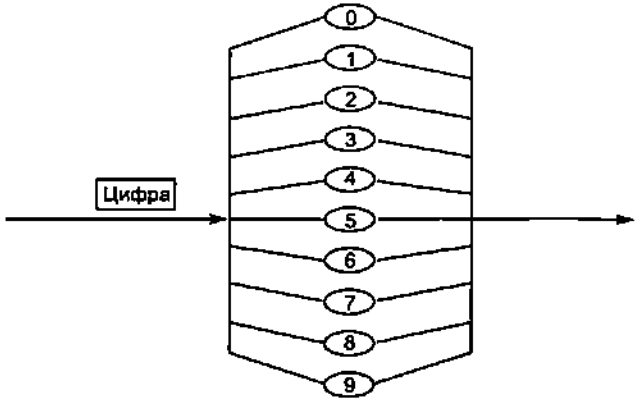


Рис. 3.9. Синтаксическая диаграмма понятия «цифра»

Синтаксические диаграммы графически изображают допустимую структуру определяемого понятия. Следование элементов структуры задается стрелками, вариантность указывается как распараллеливание направлений следования. Лексемы изображают в виде кружка или овала, а определяемое понятие — в виде прямоугольника. Диаграмма имеет одну входную и одну выходную стрелки.

Основные понятия языков программирования. Понятие *«уровень языка»* определяет уровень его абстракции. Язык низкого уровня соответствует реальному языку, который понимает процессор. Языки высокого уровня соответствуют не реальным, а некоторым гипотетическим процессорам. На языке реальной ЭВМ пишут трансляторы.

Родственные конструкции разных языков различаются главным образом внешним видом — набором ключевых слов или порядком следования компонентов. Следовательно, конструкции современных языков имеют общую семантику (содержание), но

различный порядок следования компонент (синтаксис) и разные ключевые слова (лексику).

Семантическое описание содержит, как правило, список компонент, из которых состоит конструкция (тип, имя типа), описание каждой компоненты, описание конструкции в целом. Синтаксическое описание дает формальное определение конструкции.

Большинство языков программирования оперируют одними и теми же *фундаментальными понятиями*, такими, например, как *тип данных* или *структура данных*.

Алгоритм содержит только указания на выполнение действий, а программа включает еще и некоторую дополнительную информацию. В большинство языков программирования включаются объекты двух видов: описания и предписания.

В разделе *описаний* определяются объекты, с которыми манипулирует программа (типы, структуры, имена и т. д.), задаются некоторые характеристики этих объектов. *Предписания* содержат указания на выполнение необходимых действий. Основным средством указания на выполнение некоторого действия в языках программирования является *оператор*. Операторы реализуют алгоритм решения задачи.

Как и в математике, в программировании принято оперировать *константами* и *переменными*. Есть простые переменные и переменные с индексами, которые используются для ссылки на некоторый элемент массива. Общим для большинства языков программирования является понятие *идентификатор*. Формально он определяется так:

<идентификатор> ::= <буква> | <идентификатор> <буква> | <идентификатор> <цифра>,

что означает: «идентификатор есть последовательность букв и цифр, начинающаяся с буквы». В основном идентификаторы употребляются в качестве *имен* каких-либо объектов (констант, переменных, типов, структур, программ и т. д.), что позволяет идентифицировать эти объекты.

Следующее понятие языков программирования — *«выражение»*. Оно бывает двух видов: арифметическое, логическое. Понятие *«арифметическое выражение»* в точности соответствует принятому в математике. Пример: арифметическое выражение

$$(a + b)\sin x + c^2 \operatorname{tg} y^5$$

будет записано как

$$(a + b) * \sin x + c \wedge 2 * \operatorname{tg} y \wedge 5.$$

Логическое выражение образуется с помощью знаков логических операций из объектов (констант, переменных и т. д.) буквенного типа. В логическое выражение может входить *отношение* — еще одно из фундаментальных понятий языков программирования. Например, записи a < b, c ≥ d, x ^ 2 + y ^ 2 < a ^ 2 — отношения, а запись (a < b) or (c = n) — логическое выражение.

Подпрограмма (понятие ввел американец Уилкс, 1957 г.). Понятие *«подпрограмма»* введено, чтобы избежать многократного повторения одинаковой последовательности предписаний. Подпрограмма может быть вызвана из любого места программы или другой подпрограммы с указанием тех значений исходных данных, с которыми она должна быть выполнена. Распространены подпрограмма-функция и подпрограмма-процедура.

Контрольные вопросы

1. Что такое язык программирования?
2. Каковы уровни языков программирования? Приведите примеры языков разных уровней.
3. Какие языки программирования высокого уровня были первыми?
4. Какова история развития языка программирования Бейсик?
5. Какова история развития языка программирования Паскаль?
6. Что такое система программирования? Приведите примеры.
7. Каков состав языка программирования? Дайте характеристику каждой составляющей языка.
8. Что такое компиляция программы и каковы ее основные этапы?
9. Что такое метаязык и что используют в качестве метаязыка?
10. Каковы основные понятия языков программирования?

3.4. Система программирования Visual Basic

MS Visual Basic — система программирования, позволяющая быстро и эффективно создавать приложения для ОС Windows. Система программирования Visual Basic включает инструментальную оболочку и транслятор с языка. Система является *визуальной*,

т. е. можно создавать видимую часть приложения Windows, не написав ни строки программного кода.

Разработка программного интерфейса осуществляется на принципах объектно-ориентированного подхода, поэтому для приложений характерно существование на экране в любой момент времени множества объектов: окон, кнопок, меню, текстовых и диалоговых окон, полос прокрутки и др. Пользователь свободно выбирает и использует их с помощью щелчка мыши, перетаскивания объекта, ввода данных в окно.

Процесс разработки программы в среде Visual Basic сводится к выбору набора объектов и их свойств, заданию событий и процедур их обработки, которые в совокупности обеспечивают решение поставленной задачи. Существует три варианта Visual Basic: Learning Edition (учебная редакция), Professional Edition (профессиональная редакция), Enterprise Edition (редакция для предприятий).

Запуск Visual Basic осуществляется одним из следующих способов:

- с помощью команд меню **Пуск** → **Программы** → папка **Microsoft Visual Basic** → пиктограмма **Visual Basic**;
- щелчком мыши по ярлыку **VB**;
- с помощью команд меню **Сервис** → **Макрос** → **Редактор Visual Basic**.

После запуска программы на экране появляется диалоговое окно **New Project**. В нем указан тип программного проекта (**Standard EXE**). Меню окна содержит три пункта:

- **New** — создание нового проекта (приложение VB называется проектом);
- **Existing** — выбор приложения из существующих проектов;
- **Recent** — список последних проектов.

Щелчком мыши можно выбрать нужный проект, открыть его с помощью кнопки **Open** (**Открыть**). Закрыть окно можно щелчком по кнопке **Cancel** (**Отмена**). Все функции окна **New Project** (**Новый проект**) заложены в меню **File** среды программирования, поэтому указанное диалоговое окно можно не использовать.

Среда программирования IDE (Integrated Development Environment — интегрированная среда разработки) Visual Basic показана на рис. 3.10. Главное окно содержит: строку заголовка с названием проекта, строку меню и пиктографическое меню. Меню и команды работают в соответствии со стандартными соглаше-

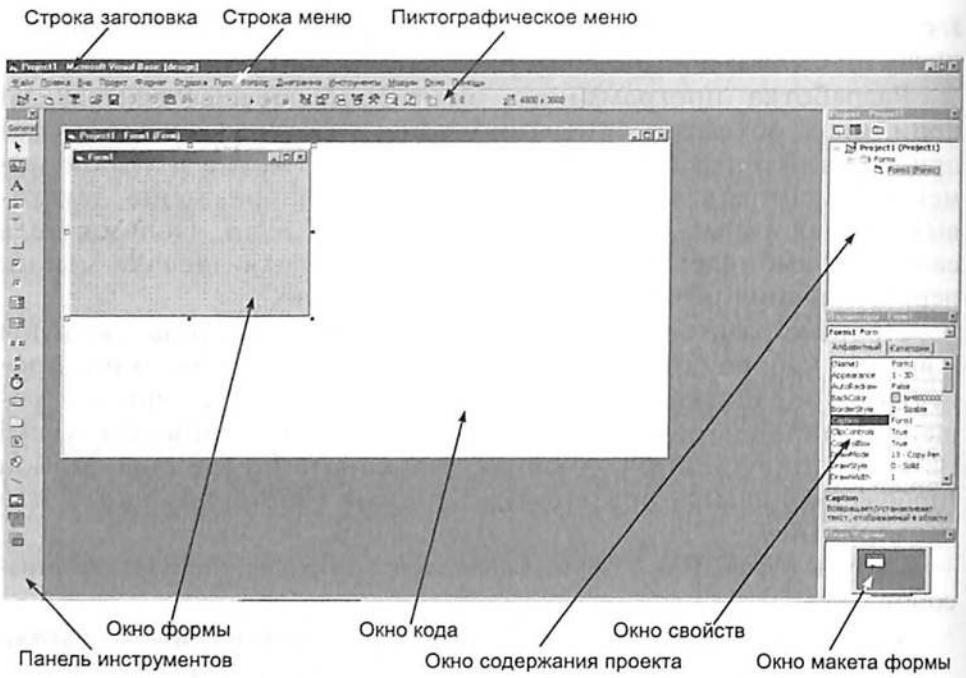


Рис. 3.10. Среда программирования IDE Visual Basic

ниями, общими для приложений Windows. Под строкой меню расположена горизонтальная панель, представляющая собой набор кнопок, являющихся ярлыками для команд, с помощью которых осуществляется работа в среде Visual Basic. В среде имеется семь окон, которые используют для разработки приложений.

Под панелью пиктограмм слева направо располагаются:

1. Панель инструментов, содержащая различные объекты (командные кнопки, текстовые окна, ярлыки и др.) — элементы управления, которые перетаскиваются на форму и становятся объектами или программируемыми элементами пользовательского интерфейса. Имеются также средства управления для создания спецопераций по управлению информацией в БД, контроля интервалов времени и др. Для расширения средств управления панели в меню **Project (Проект)** имеется пункт **Components (Компоненты)**.

2. Окна формы **Form** и **Project1 — Form1 (Form)**, находящиеся внутри окна проекта. *Форма* — это окно интерфейса пользователя. На форме располагают меню, кнопки, окна списков, полосы прокрутки и др. В начале разработки среда программирова-

ния предлагает одну форму **Form1** с регулярной стандартной сеткой, с помощью которой размещают элементы интерфейса. Можно добавлять новые формы командой **Add Form (Добавить форму)** в меню **Project**. Для каждой формы создается свой файл. Файл формы имеет расширение .frm.

3. Окно кода. Добавить в проект новый модуль можно, выполнив команды **Project → Add Module (Проект → Добавить модуль)**. В редакторе кода осуществляется программирование задач. Файл кода имеет расширение .bas.

4. Окно содержания проекта **Project (Project Container)** предназначено для работы с отдельными компонентами проекта. Проект содержит файлы форм, кода программы, исполняемый файл и дополнительные файлы, использующие, как правило, инструменты не из Visual Basic.Файл проекта имеет расширение .vbp и содержит список всех файлов проекта. Его структура отображается в окне в виде дерева. Открыть окно проекта можно, выбрав кнопку **Project Explorer (Проводник проекта)** на панели инструментов.

5. Под окном содержания проекта обычно находится окно свойств **Properties (Параметры)**. Каждый объект имеет набор свойств, которые определяют внешний вид и поведение объектов. С помощью этого окна их можно устанавливать или изменять. Свойства могут быть числовыми, булевыми (истина или ложь) и текстовыми.

6. Окно **Form Layout (План-макет Формы)** предназначено для установки (путем перемещения) начального положения форм на экране.

7. Окно непосредственного выполнения **Immediate** находится ниже окна инструментов и окна содержания проекта.

Расположение и форму окон можно изменять, а также сворачивать их, чтобы сделать доступными и видимыми на экране необходимые элементы среды программирования.

Получение справочной информации в Visual Basic осуществляется несколькими способами:

- щелкнуть по кнопке **Help (Справка)** в диалоговом окне или нажать на клавишу **F1** — для получения информации о работе в окне;
- в меню **Help** выбрать команду **MS Visual Basic Help Topics (Справка по MS Visual Basic)**, далее выбрать вкладку **Contents (Содержание)** и нужную тему — для получения информации по предмету или операции;

- в окне **Help Topics (Справка)** выбрать вкладку **Index (Предметный указатель)** и в поле ввода **Type the first few letters of the Word you're looking for (Введите ключевые слова)** ввести название темы, из списка выбрать термин и щелкнуть по кнопке **Display (Найти)** — для получения справки по конкретному вопросу об инструментах программирования, свойствах или элементах языка.

Чтобы закрыть окно Visual Basic, следует в главном окне щелкнуть по кнопке **Свернуть**.

Для того чтобы выйти из программы, нужно в меню **File (Файл)** выбрать команду **Exit (Выход)**.

Если проект сохраняется впервые, появится диалоговое окно, в котором нужно выбрать папку для сохранения проекта, набрать имя файла и щелкнуть по кнопке **Save (Сохранить)**. Visual Basic сначала сохранит форму, а затем появится диалоговое окно **Save Project AS (Сохранение проекта)**. Нужно поступить так же, как в первый раз. В результате произойдет сохранение проекта на диске в файле с расширением .vbp, после чего Visual Basic закроется. Если необходимо не сохранять изменения и продолжить работу в Visual Basic, то следует щелкнуть по кнопке **Cancel (Отмена)**.

Большинство объектов, которые создаются, уже «знают», как работать после запуска программы, и готовы к приему данных. Внутренняя функциональность создаваемых объектов — одна из мощнейших особенностей Visual Basic. В разрабатываемой программе недостает только кода, который будет выполнять предлагаемые пользователем действия. Вычислительная логика может быть встроена в приложение с помощью программного кода. Например, если действия управляются кнопками, то для работы с программным кодом нужно дважды щелкнуть мышью по соответствующей командной кнопке. После этого появляется окно **Code (Код)**. Блок кода, связанный с объектом интерфейса, называют *процедурой события* Visual Basic. Тело процедуры заключено между операторами, указывающими на начало и конец подпрограммы:

```
Private Sub Camnandl_ Click
End Sub
```

Операторы тела и процедуры выполняются каждый раз, когда пользователь активизирует элемент интерфейса, ассоциированный с процедурой. Следует иметь в виду, что у каждого объекта интерфейса может быть несколько ассоциированных с ним

процедур — по одной для каждого связанного с этим объектом события.

Visual Basic содержит развитые средства контекстной оперативной помощи пользователю при наборе программного кода. При наборе пользователем утверждения в окне **Code** появляется черный текст. После набора строки и нажатия клавиши **Enter** цвет утверждения изменяется на синий, текст форматируется: строчные буквы в начале ключевых строк заменяются заглавными, где нужно, расставляются пробелы. Это говорит о том, что Visual Basic распознал строку как допустимую. Когда пользователь начинает набирать свойство, Visual Basic показывает доступные для объекта свойства в списке, так что пользователь может дважды щелкнуть мышью по свойству вместо того, чтобы его набирать. Кроме того, Visual Basic выделяет цветом различные фрагменты текста, чтобы помочь их идентифицировать, а также ошибки. После завершения работы над программой ее нужно сохранить.

Особенности Visual Basic как объектно-ориентированного языка программирования заключаются в следующем. В частном случае в Visual Basic *объектом* называется элемент пользовательского интерфейса, который создается на форме. Каждый объект есть представитель некоторого *класса* однотипных объектов, и объект является экземпляром класса. Класс определяет общие для всех его объектов методы и свойства. *Методы* — это программные процедуры, определяющие взаимодействие объектов класса с внешней средой. *Свойства* — это характеристики (атрибуты) объектов (название, цвет, размер шрифта и др.).

Инкапсуляция проявляется в том, что доступ к объекту скрыт и возможен только через его методы и свойства. Объекты — это структуры, отделенные от внешней среды. Инкапсуляция позволяет изменять реализацию объектов любого класса без опасения повредить программную систему. Это обеспечивает многократное использование одного и того же программного кода.

Полиморфизм — это способность объектов выбирать операцию на основе данных, принимаемых в сообщении. Каждый объект может реагировать по-своему на одно и то же сообщение. Например, команда **Print** (**Печать**) будет по-разному воспринята черно-белым или цветным принтером.

Наследование — это возможность выделить свойства, методы и события одного объекта и передать их другому объекту, иногда с модификацией. С точки зрения программиста, новый класс должен содержать только коды и данные для новых или изме-

няющихся методов. Недостаточные возможности Visual Basic по поддержке наследования не позволяют считать его истинно объектно-ориентированным языком программирования, как Си++, Delphi (Object Pascal) и Java.

Объекты, методы, свойства, события. Объекты Visual Basic (формы и элементы управления и отображения на них) функциональны. При этом если свойства объекта определяют его внешний вид и поведение, то методы объекта определяют задачи, которые может выполнить данный объект. Методы представляют собой сегмент программного кода, внедренный в объект. Существует *определенный формат программного кода*, задающего установку свойства и использование метода:

```
Объект. Свойство = Значение
Объект. Метод [Параметр1 [...] ],
```

где `Объект` — имя настраиваемого объекта; `Свойство` — характеристика, которую нужно изменить; `Метод` — команда, которая используется для изменения объекта; `Значение` — новая установка свойства; `Параметр` — аргумент, используемый методом.

Разные объекты обладают различными методами. Наиболее часто используемые методы (см. раздел 3.6): `Drag`, `Move`, `SetFocus`, `Zorder`.

Объекты могут реагировать на события (например, на щелчок мыши по кнопке команды, изменение текста и др.). Каждый объект генерирует свои события. Некоторые события (например, `Click`) являются общими для многих типов элементов управления. Хотя, например, `Click` для формы отличается от события `Click` для кнопки команды. Иногда выполнение некоторого метода приводит к изменению свойств объекта, а изменение некоторых свойств может вызвать наступление событий. Например, метод `Move` (Переместить) приводит к изменению свойств `Left` и `Top`, определяющих положение объекта относительно верхнего левого угла формы. Изменение свойств `Height` или `Width` формы с помощью кода вызывает событие `Resize` формы и т. д.

Windows автоматически распознает события и при каждом событии посылает программе сообщение. Программа должна интерпретировать полученное сообщение, т. е. определить, какое событие стоит за этим сообщением, и выполнить соответствующие действия. Программа может обрабатывать два основных типа событий: инициируемые пользователем и генерируемые системой.

События, инициируемые пользователем, возникают в результате его действий: нажатия клавиш, щелчков кнопками мыши. Но есть события, являющиеся следствием действий пользователя. Например, когда пользователь щелкает мышью в поле текста, чтобы приступить к редактированию, для объекта `TextBox` вызывается событие `Click`. Но вместе с ним вызывается еще несколько событий: `GotFocus` (Получить фокус) для `TextBox` и `LostFocus` (Потеря фокуса) для соответствующего элемента управления.

Основными действиями пользователя, вызывающими события в программе, являются следующие: запуск программы; нажатие клавиши; щелчок кнопкой мыши; перемещение мыши; выход из программы. Для многих элементов управления общими являются следующие (см. раздел 3.6): `Change`, `Click`, `Dbclick`, `DragDrop`, `DragOver`, `GotFocus`, `KeyDown`, `Keypress`, `KeyUp`, `LostFocus`, `MouseDown`, `MouseMove`, `MouseUp`.

Программирование, управляемое событиями. Каждый элемент управления или форма отвечает за определенный набор событий, но совсем необязательно обрабатывать их все. Нужно написать код для какого-либо события. Для этого нужно выбрать объект и его событие из раскрывающихся списков вверху окна редактирования кода той формы, в которую помещен этот объект. Сначала нужно выбрать объект из левого списка, а затем интересующее событие — из правого. Visual Basic покажет пустую процедуру для выбранной комбинации объект/событие, в которой и нужно написать код.

Имя процедуры обработки события содержит имя объекта и имя события, разделенные символом подчеркивания.

Структура проекта. Управление проектами. Простые проекты в Visual Basic строятся из форм, модулей и элементов управления ActiveX. *Модуль* — это файл, содержащий код, не связанный ни с одной формой или элементом управления. Модули содержат исключительно программные инструкции: объявления переменных, определение констант, функции и подпрограммы, определяемые пользователем. Файл, хранящий форму, имеет расширение .frm, а файл модуля — .bas. Файлы элементов управления ActiveX имеют расширение .ocx.

Для упрощения работы с составными частями проекта имеется панель **Project Explorer (Проводник проекта)**, которая дает представление проекта в виде дерева форм и модулей, позволяет просмотреть любой файл, модуль или форму в окне редактирования кода, а также имеет удобное контекстное меню.

Характеристики приложения даны в специальном объекте App, который имеет набор свойств. Устанавливаются свойства в диалоговом окне **Project Properties (Свойства проекта)** проекта. Работа со свойствами объекта App важна при создании программы, так как с их помощью можно следить за версиями программы, соблюдением авторского права и др.

При сохранении проекта (включая все формы и модули) создается файл проекта с расширением .vbp.

Компиляция проекта. После установления свойств объекта App проекта выполняется компиляция в одном из двух форматов: P-Code (Псевдокод), Native Code (Собственный код). В первом случае исполняемый файл запускается как интерпретируемый код, т. е. программа во время выполнения должна считывать информацию из библиотек динамической компоновки (DLL) программы. Если же программа компилируется в собственный код, то файлы проекта преобразуются в более эффективный двоичный код, который выполняется быстрее. Но собственный код все равно требует подключений библиотек DLL во время выполнения программы. Разница заключается в том, что доступ к библиотекам и их использование осуществляются с помощью исполняемых файлов (.exe) по-разному. Исполняемый файл в формате собственного кода обычно имеет больший размер, чем файлы в формате псевдокода. Для того чтобы провести компиляцию проекта в стандартный EXE-файл, нужно выполнить следующие действия:

1. Открыть проект, предназначенный для компиляции.
2. В меню **File (Файл)** выбрать команду **Make Имя_Проекта.exe**. Откроется диалоговое окно **Make Project (Сборка проекта)**.
3. При необходимости переименовать файл в поле **Filename (Имя файла)**.
4. Щелкнуть по кнопке **Options (Параметры)**, чтобы открыть диалоговое окно **Project Properties (Свойства проекта)**. На вкладке **Compile (Компиляция)** выбрать тип компиляции.
5. Щелкнуть по кнопке **OK**, чтобы закрыть окно **Project Properties**, а затем — по кнопке **OK** в окне **Make Project**, чтобы начать компиляцию программы.

В результате будет получен исполняемый файл, который можно будет запускать вне среды IDE Visual Basic.

Отладка программ. Назначение отладки — определить местоположение ошибок и устранить их. Ошибки бывают трех видов: синтаксические; логические; при выполнении программы. Но

самое главное при написании программы — придерживаться принципов защиты от ошибок, для чего существует несколько проверенных способов. Ошибки начинающего программиста связаны, как правило, с плохим знанием элементов языка. Профессиональный программист обычно не допускает ошибок на этапе кодирования, он может ошибаться на начальных этапах работы, когда программа только проектируется и не все детали алгоритма до конца ясны.

Синтаксические ошибки — это ошибки в структуре кода (в написании, грамматике или пунктуации). Синтаксические ошибки обнаруживаются компилятором автоматически.

Логические ошибки — это ошибки в логике программы или в значениях переменных.

Ошибки выполнения — это ошибки, возникающие в процессе выполнения программы. Самые распространенные ошибки может выявлять сама программа.

Синтаксические ошибки можно зачастую устранить, проверив, все ли переменные объявлены. Логические ошибки идентифицируют с помощью точек останова (контрольных точек) и пошагового выполнения программы. Основные средства отладки можно получить через меню **Debug (Отладить)** и панель инструментов, вызвать которую можно командами **View (Вид)** → **Toolbars (Панели инструментов)** → **Debug**. Сообщения о найденных ошибках отображаются в окне кода, где подсвечивается строка, в которой эта ошибка обнаружена. Ошибки исправляют и запускают программу на дальнейшую компиляцию (**Run** или **F5**). Отлаживать программу можно с помощью создания *точек останова* разными способами (например, **Debug** → **Toggle Breakpoint (Установить точку останова)**). Точки останова используют для ограничения области поиска ошибок в программном коде. Более подробно наблюдать за данными программы можно с помощью *просматриваемых значений* (окно **Watches (Наблюдаемые переменные)**).

Создание дистрибутива приложения. Для передачи приложения на другой компьютер нужно создать дистрибутив приложения. Это можно осуществить с помощью Мастера программы установки — Application Setup Wizard. Нужно выполнить следующие действия:

- запустить программу Application Setup Wizard. Появится диалоговое окно;

- нажать кнопку **Next (Далее)**, а затем выбрать опцию **Create a Setup Program (Создать программу установки)**. Откроется новое диалоговое окно;
- нажать кнопку **Browse (Обзор)** для поиска файла проекта (файла с расширением .vbp). Далее щелкнуть по кнопке **Next**;
- в окне **Distribution Method (Способ размещения)** выбрать параметр **Floppy Disk** (если приложение будет размещено на дискетах) или параметр **Single Directory** (если приложение будет размещено на компакт-диске или сетевом диске). Если проект большой, его можно занести в несколько каталогов с образами дисков (Disk1 \ Disk2...), которые потом можно записать на CD. Далее щелкнуть по кнопке **Next**;
- в диалоговом окне **Floppy Disk** указать имя дисковода (A или B) и емкость дискеты. Нажать кнопку **Next**. Если использовались компоненты ActiveX, то их нужно указать в окне **ActiveX Server Components**;
- в окне **File Summary** должны присутствовать все необходимые для проекта файлы, иначе следует, используя кнопку **Add (Добавить)**, добавить недостающие файлы и щелкнуть по кнопке **Next**;
- для сохранения записи процесса создания программы нужно щелкнуть по кнопке **Save Template (Сохранить шаблон)**. Сохраняется файл с расширением .swt;
- щелкнуть по кнопке **Finish (Готово)**. Появится окно копирования файлов **Working**, которые нужны для установки приложения, а затем — окно с указанием количества требуемых дискет. Щелкнуть по кнопке **OK**;
- выполнить просьбы об установке последовательности дискет;
- после выполнения процесса появится сообщение о создании программы установки. На первой дискете находится файл Setup.exe. Компонент ActiveX и библиотеки динамической компоновки находятся в каталоге Windows/System. Программа Setup.exe в реестре ОС создает все необходимые записи. В меню **Пуск** добавится строка с созданным приложением.

Контрольные вопросы

1. Каково назначение и состав системы MS Visual Basic?
2. Что такое среда программирования IDE Visual Basic?

3. Как называются окна среды программирования IDE Visual Basic?
4. В чем заключается процесс разработки программы в среде Visual Basic?
5. Как получить справочную информацию в среде Visual Basic?
6. Что такое свойства объектов, события и методы?
7. Что такое инкапсуляция, полиморфизм и наследование?
8. Каков формат программного кода для установки свойства и использования метода?
9. Какова структура проекта в Visual Basic?
10. Как выполняется компиляция проекта?
11. Какие ошибки возникают при отладке программ?
12. Как создать дистрибутив приложения?

3.5. Язык программирования Visual Basic

Для создания программных модулей служат средства языка *Visual Basic*. Язык имеет следующий алфавит:

- прописные и строчные буквы латинского алфавита от A до Z;
- прописные и строчные русские буквы от А до Я;
- арабские цифры от 0 до 9 (используются для записи чисел и наименований различных объектов программы;
- символ подчеркивания _;
- неотображаемые символы, используемые для отделения лексем друг от друга — пробел, табуляция, переход на новую строку;
- знаки арифметических операций: + (сложение), - (вычитание), * (умножение), / (деление), ^ (возведение в степень);
- знаки операций отношений: = (равно), > (больше), < (меньше), >= (больше или равно), <= (меньше или равно), <> (не равно);
- разделители и прочие символы: . (точка), , (запятая), ; (точка с запятой), : (двоеточие), ?, ! (вопросительный и восклицательный знаки), % (процент, признак целой величины), ¤ (признак текстовой величины), \ (разделитель операторов), () (скобки круглые), [ ] (скобки квадратные), ' (апостроф), " (кавычки), # (номер), & (логическое И), @ и др.

Программный код — последовательность лексических единиц (лексем), записанная по синтаксическим правилам языка и реализующая некоторую семантическую конструкцию.

Комментарий — пояснительный текст, представляющий собой последовательность символов, размещаемых на одной строке исходного текста программы, начинается с апострофа ' или ключевого слова Rem.

Лексема — единица текста программы, имеющая смысл для компилятора. Различают шесть классов лексем: идентификаторы, служебные (зарезервированные) слова, константы, строки (строковые константы), операции (знаки операций), разделители (знаки пунктуации).

Зарезервированные слова: All, As, ASC, Binary, BY, ByRef, ByVal, CREATE, Date, DESCDROP, Else, Empty, Error, False, For, Friend, Get, IN, INDEX, Input, INTO, Is, JOIN, Len, Lock, Me, Mid, New, Next, Nothing, Null, ON, On, Option, Optional, ParamArray, Print, Private, Property, Public, Resume, Seek, SELECT, Set, Static, Step, String, TABLE, Then, Time, To, WITH, WithEvents.

Идентификатор — последовательность букв, цифр и символов подчеркивания для обозначения имен переменных, констант, процедур и функций.

Имя должно начинаться с буквы, не должно содержать точек, пробелов, разделительных символов, знаков операций, специальных символов, иметь длину не более 255 символов (VB учитывает только первые 31). Имена могут быть простыми и составными. Например, Number — простое имя переменной, ФамилияСтудента — составное.

Переменные — объекты для хранения данных. В разные моменты могут хранить разные значения. На логическом уровне — имя переменной, на физическом уровне — область памяти. При объявлении переменной указывается ее имя и тип.

Типы данных: число, строка текста, экземпляр объекта, элементы управления, базы данных и др. Различают две группы типов данных: основные (базовые) и определяемые пользователем. Базовые типы:

- *целочисленные*: Byte (1 байт, от 0 до 255); Boolen — логический (2 байта, True или False); Integer (2 байта, от –32 768 до 32 767); LongInteger (4 байта, +/–2.1E9);
- *с плавающей точкой:* Single (вещественные числа одинарной точности, 4 байта); Double (вещественные числа двойной точности, 8 байт); String (строка фиксированной длины, 1 байт на каждый символ текста, от 1 до 65 400);

`String` (строка переменной длины, 10 байт + 1 байт на каждый символ текста, от 0 до 2 млрд символов);

- *объектные типы:* объект (рисунок или ссылка на любой другой объект, 4 байта);

  `Variant`: значения любого из перечисленных типов данных (16 байт для чисел, 22 байта + 1 байт на каждый символ для строк);
- *прочие:* `Currency` — денежные единицы (8 байт, 15 цифр до десятичной точки и 4 — после); `Date` — информация о дате и времени (8 байт, от 1.01.100 г. до 31.12.9999 г.); `Decimal` — десятичное число (14 байт, целое — 29 знаков, вещественное — 27 знаков после запятой).

Декларация переменных может быть явной или неявной. Явное определение переменных выполняется двумя способами.

Способ 1 предполагает использование следующего синтаксиса:

```
Static | Public | Private Dim Имя Переменной1
[As Тип1], Имя Переменной2 [ As Тип2]]
```

где `Dim` (Размер) — ключевое слово, которое сообщает VBasic, что декларируется переменная и резервируется область памяти для ее хранения.

Для описания переменных в VBA чаще других используется служебное слово `Dim`. Например, `Dim A As Single` (A описывается как переменная типа `Single`); `Dim D` (переменная `D` будет типа `Variant`).

При описании константы ей должно быть присвоено значение. Например, `Const Pi as Double = 3.14159`.

Константы цветов для окраски фона или шрифта: красный — `Const =VbRed`; зеленый — `Const = VbGreen`; желтый — `Const = = VbYellow`; синий — `Const = VbBlue`; розовый — `Const = = VbMagenta`; голубой — `Const = VbCyan`; черный — `Const = = VbBlack`; белый — `Const = VbWhite`. Описание стандартных констант приведено в справочной системе VBA:

- `Имя Переменной` — имя переменной (идентификатор, не входит в перечень ключевых слов);
- `As` (Как) — ключевое слово, которое сообщает, что определяется тип данных для переменной;
- `Тип` — тип данных для объявляемой переменной;

- Private (Частный), Public (Общий) — ключевые слова, определяющие область видимости переменной;
- Static (Статический) — ключевое слово, определяющее, сохраняет ли переменная свое значение при завершении блока программы (процедуры, функции) и выходе из него.

Служебное слово Public используется для описания общих переменных на уровне модуля. Например, Public ObName As String. Доступны для процедур всего проекта.

Служебное слово Private используется для описания «личных» переменных на уровне модуля. Например, Private LName As String. Доступны для процедур одного модуля.

Служебное слово Static используется для сохранения значений переменных при выполнении программы.

Способ 2 предусматривает указание типа с помощью суффикса (символа типа). В этом случае тип данных при объявлении переменных имеет следующий синтаксис:

```
Static | Public | Private Dim Имя ПеременнойСуффикс
```

Например, Dim strInputMag$ — объявляется переменная типа String (Строка).

Типы переменных и соответствующие им суффиксы, применяемые при декларации типов, перечислены в табл. 3.2.

Неявное объявление переменных осуществляется также двумя способами.

Способ 1 заключается в использовании оператора DefТип_Данных. Этот оператор устанавливает тип данных для переменных, параметров процедур и тип возвращаемого значения для процедур типа Function и Property Get, имена которых начинаются с определенных символов. Синтаксис оператора имеет следующий вид:

```
DefТип_Данных ДиапазонБукв [ , ДиапазонБукв ] ...
```

где Тип_Данных — сокращенное название типа данных;

ДиапазонБукв — указывает границы диапазона имен, для которых задается тип данных по умолчанию.

Например,

```
DefInt A-K                    'Переменные, имена которых
'начинаются с символов A-K, будут иметь тип Integer.
CalcVar = 4                   'Инициализация целого числа.
StringVar ="Привет вам"   'Инициализация строки.
```

Таблица 3.2. **Типы переменных и соответствующие им символы**

Тип переменной	Символ типа
`Integer`	%
`Long`	&
`Single`	!
`Double`	#
`Currency`	@
`String`	$
`Byte`	Нет
`Boolean`	Нет
`Date`	Нет
`Object`	Нет
`Variant`	Нет

Способ 2 допускает просто указание имени переменной, например `MyVal` или `MyNameS`, в тексте программы. Хороший стиль программирования предполагает явную декларацию с помощью ключевых слов `Dim`, `Private`, `Public`, `Static`.

Строковые переменные. Строки могут быть переменной и фиксированной длины. Строки переменной длины могут содержать до 2 млрд символов. Строка фиксированной длины — это строка постоянного размера, указанного при объявлении переменной. Если такой строке присваивается значение более длинное, то лишние символы отбрасываются. Если значение, которое присваивается, короче, то остающееся место заполняется пробелами. Синтаксис декларации следующий:

```
Dim VarName As String * ДлинаСтроки,
```

где `ДлинаСтроки` — целочисленная переменная или константа, которая содержит число, указывающее длину строковой переменной.

Константы могут быть именованными и неименованными. Синтаксис языка различает три типа констант: символьные, целые и вещественные.

Перечни служат для объявления группы констант под общим именем и принадлежащих типу `Enumeration` (Перечень). Объ-

является только в разделе глобальных объявлений модуля или формы. Синтаксис следующий:

```
[Public / Private] Enum Имя Перечня
Имя константы 1 [=значение]
Имя константы 2 [=значение]
Имя константы N [=значение]
End Enum
```

Синтаксис для обращения к константам перечня следующий:

```
Имя Перечня. Имя Константы
```

Область видимости (доступность) переменных и констант может быть как во всей программе, так и в ее отдельных частях. Область видимости переменной декларируется одним из ключевых слов:

- `Dim` — объявляет локальные переменные, существующие только во время вызова процедур или функций, в которых они объявлены. Если переменная объявляется как глобальная модуля или формы, то она доступна для всех процедур и функций только этого модуля;
- `Private` — не может объявлять переменную внутри процедуры или функции, при объявлении в разделе глобальных объявлений модуля `Dim` и `Private` равнозначны;
- `Public` — объявляется как глобальная переменная на уровне приложения, доступна из всех его модулей.

Статические переменные объявляются внутри процедуры или функции и вне их недоступны. Объявляются как `Static`.

Массивы — переменные структурного типа, предназначенные для идентификации и резервирования памяти для нескольких однотипных данных. Массивы могут быть одномерные и многомерные.

Объявление массива имеет следующий вид:

```
Dim | Public | Private Имя_Массива (Индексы)_
As Тип_Данных
```

где `Dim`, `Public`, `Private` — ключевые слова, декларирующие массив и область его видимости;

`Имя_Массива` – идентификатор имени массива;

`Индексы` — значение индекса (номера) последнего элемента в массиве, считая с нулевого;

As — ключевое слово, предваряющее указание типа элементов массива;

Тип_Данных — любой, действительный для Visual Basic тип данных (базовый или созданный пользователем). Например,

```
Dim MyArray (7) As Integer          'Одномерный массив
                                    'из восьми элементов
Dim strMyArray (4,5) As String      'Двумерный массив
                                   ' из 5 строк и 6 столбцов.
```

В скобках указана размерность массива, базовый индекс массива равен 0.

Но если в раздел "General" поместить оператор Option Base 1, то первый индекс будет равен 1. Базовый индекс можно изменить, используя при объявлении массива ключевое слово To. Например, DIM B (2 To 7) As String. Здесь базовый индекс массива равен 2.

Если в процессе выполнения программы размер массива нужно изменить, то массив объявляется как динамический. Для этого в объявлении не указывается размерность, например:

```
Dim B () As String.
```

Количество элементов в объявленном массиве и его размерность в процессе выполнения программы можно переопределить с помощью ключевого слова ReDim. Синтаксис переопределения массива имеет следующий вид:

```
ReDim [Preserve] ИмяМассива (индексы) [As ТипДанных],
```

где ReDim — ключевое слово, указывающее, что переопределяются размеры массива;

Preserve — необязательное ключевое слово, с помощью которого дается указание, чтобы все элементы переопределяемого массива сохранили свое значение;

Индексы — размерности массива (до 60).

Индекс массива может быть задан не только числом, но и выражением, например A(i+1). Это значит, что для работы с элементами массива можно использовать циклы. Начальным значением цикла в этом случае будет не 1, а 0. Базовые индексы могут быть и отрицательные.

Типы данных, определяемые пользователем. Они являются типами структурного вида и создаются на основе базовых типов Visual Basic. Создавать свои типы данных полезно тогда, когда

программа работает с группой элементов различного базового типа, но связанных между собой по смыслу.

Создание нового типа осуществляется следующей конструкцией:

```
Type ИмяТипа
Имя1 As Тип      'Структурный элемент создаваемого типа -
                 'базовый тип
Имя2 As Тип      'Структурный элемент создаваемого типа -
                 'базовый тип
...
ИмяN. As Тип     'Структурный элемент создаваемого типа -
                 'базовый тип
End Type
```

где Type — ключевое слово, которое указывает, что создается новый пользовательский тип данных;

ИмяТипа — имя создаваемого типа (идентификатор);

ИмяN As Тип — описание структурного элемента создаваемого типа;

End Type — ключевые слова, завершающие описание нового типа.

Например:

```
Type Сотрудник
strФамилия As String       ' Структурный элемент для
                           ' хранения фамилии
ДатаРождения As Date       ' Структурный элемент для
                           ' хранения даты рождения
ДатаПоступления As Date    ' Структурный элемент для
                           ' хранения даты поступления
                           ' на работу
End Type
```

Для использования созданного типа данных программе надо в разделе объявлений объявить переменную такого типа. Объявление переменной такое же, как и в случае базовых типов:

```
Dim udtСлужащий As Сотрудник ' Объявлена переменная
                             'пользовательского типа Сотрудник
```

Обращение к элементу пользовательского типа имеет следующий синтаксис:

```
Имя Переменной.ИмяСтруктурного Элемента
```

Например:

```
udtСлужащий.strФамилия="Иванов"
```

Этот тип данных может содержать структурные элементы, тип которых также является пользовательским, например массив.

Операторы, выражения и операции. *Программный оператор* — строка с кодом в исходном тексте программы, предложение, выполняющее какое-либо действие. Это может быть комбинация ключевых слов Visual Basic, функций, операций и символов. Такую конструкцию должен распознавать компилятор Visual Basic. Оператор может быть единственным ключевым словом или состоять из комбинации элементов. Операторы строятся по определенным правилам, синтаксису. Оператор может включать *выражения* (Expression) — комбинацию знаков операций и операндов, а также скобки. Назначение любого выражения — получение некоторого значения. Синтаксическая конструкция выражения:

```
Операнд1 [операция Операнд 2 [операция Выражение]]
```

Тип формируемых значений определяет тип выражения. Например, если значениями выражения являются целые и вещественные числа, то это будут *арифметические выражения*. Операции служат для формирования и вычисления выражений. Операнды связаны между собой знаками операций, которые компилятор воспринимает как отдельные лексемы. Каждая операция имеет свой ранг (приоритет). Скобки означают операции с приоритетами.

Операция присваивания используется для присвоения переменной нужного значения и имеет следующий синтаксис:

```
Имя Переменной = Выражение
```

Например:

```
intK = 7                  ' Переменной intK целого типа
                          ' присваивается значение 7
Рабочий. Фамилия = "Кузнецов" ' Переменной
                              ' пользовательского типа
                              ' (ее структурной
                              ' составляющей Фамилия)
                              ' присваивается фамилия
                              ' Кузнецов
```

Математические операции используются для записи формул. Основными являются операции: сложения (+, ранг 7); вычитания (-, ранг 7); умножения (*, ранг 4) и деления (/, ранг 4). Остальные операции называются дополнительными: изменение знака числа (-, ранг 3); целая часть от деления (\, ранг 5); остаток от деления (Mod, ранг 6); возведение в степень (^, ранг 2). Синтаксис применения математических операций следующий:

```
Result = Операнд1 Операция Операнд2 [Операция Операнд3...
 [ ... Операция ОперандN]
```

Операции отношения используются для записи выражений условия и могут быть присвоены переменным типа `Boolean` или свойству объекта. В результате операция может возвращать только значения `True` или `False`. Знаки операций отношений: = (равно), > (больше), < (меньше), >= (больше или равно), <= (меньше или равно), <>(не равно). Кроме того, могут использоваться операции `[Строка] Like [Маска]` — соответствие строки маске — и `[Операнд1] Is [Операнд2]` — ссылка на объект (если обе переменные ссылаются на один и тот же объект, то возвращается `True`).

Логические операции используются в логических выражениях. Знаки логических операций:

- `And` — если связываемые операнды (условия) истинны, то результат будет `True`;
- `Or` — если один из связываемых операндов (условия) `True`, то результат будет `True`;
- `Not` — если условие имеет значение `True`, то результат будет `False`, и наоборот;
- `Xor` — если только одно условие имеет значение `True`, то результат будет `True`; если оба связываемых условия имеют одинаковые значения, то результат будет `False`.

Строковая операция только одна — конкатенация, которая позволяет объединить значения нескольких строковых переменных или строковых констант. Знак операции — & (амперсанд). Результатом операции является длинная строка, в которой вторая строка добавлена в конец первой. Синтаксис выражения:

```
strВыражение_1 & strВыражение_2 [ . . . & strВыражение_N ]
```

Для объединения строк можно также использовать операцию сложения +.

Операторы передачи управления применяются (но лучше их избегать) для выполнения безусловных конструкций. Синтаксис:

```
GoTo идентификатор
```

где `GoTo` — ключевое слово перехода;

`идентификатор` — метка, помещаемая слева от оператора и отделяемая от него двоеточием.

Например:

```
m1: Text9.Text = «Метка»
GoTo m1
```

Операторы ветвления (*выбора*) применяются для выполнения условных конструкций.

1. *Линейный оператор* `If . . . Then` используется для выполнения одного оператора, если условие (оцениваемое выражение, функция) будет истинным.

Синтаксис:

- безальтернативная форма: `If <условие> Then <Выражение>`;
- альтернативная форма: `If <условие> Then < Выражение_1> Else < Выражение_2>`.

2. *Блочный оператор* `If . . . Then` используется для выполнения блока операторов, который может быть как безальтернативным, так и альтернативным.

Структура безальтернативной формы:

```
If условие Then
     Программный оператор 1
     Программный оператор 2
     ...
     Программный оператор n
End If
```

Структура альтернативной формы:

```
If условие Then
     Блок программных операторов, выполняемых при значении
     условия True
Else
     Блок программных операторов, выполняемых при значении
     условия False
End If
```

3. Операторы `If` могут быть *вложенными*. Применяются, когда нужно проверить условие при наличии другого условия, являющегося истинным.

Структура:

```
If условие_1 Then
     If условие_2 Then
     Блок операторов 1
Else
     Блок операторов 2
     End If
End If
```

4. Конструкция `If ... Then ... ElseIf` для работы с *несколькими операторами* `If`. Применяется, когда есть несколько условий в дополнение к исходному.

Синтаксис:

```
If выражение_условия 1 Then
     Программный оператор 1
ElseIf выражение_условия 2 Then
     Программный оператор 2
          Else
     Программный оператор 3
End If
```

5. *Переключатели* реализованы оператором `Select Case`. Он позволяет сделать выбор из нескольких вариантов в зависимости от условия.

Синтаксис:

```
Select Case Выражение
Case Значение_1
     Оператор, выполняемый при совпадении Значения_1
     и значения Выражения
Case Значение_2
     Оператор, выполняемый при совпадении Значения_2
     и значения Выражения
. . .
Case Значение_N
     Оператор, выполняемый при совпадении Значения_N
     и значения Выражения
End Select
```

В этом операторе можно использовать операции отношения путем включения в выражение ключевых слов `To` (задает диапазон значений) и `Is` (для сравнения значения проверяемой переменной со значением выражения, следующим за словом `Is`).

Операторы циклов применяются для выполнения многократно повторяемых операторов:

1. *Циклы со счетчиком* (с известным числом повторений) реализованы как циклы `For` или `For ... Next`.

Синтаксис:

```
For Счетчик =Начальное значение To Конечное значение
[Step Шаг]
операторы
[Exit For]
Next [Счетчик цикла]
```

Например,

```
Dim I As Integer
For I = 1 To 4
     Sum =Sum + 2
Next I
```

Цикл `For Each` — специфическая форма цикла `For`. Применяют для операции с каждым объектом, входящим в группу объектов.

Синтаксис:

```
For Each Имя объекта In Имя коллекции
Код операций над объектами
Next Имя объекта
```

2. *Циклы с условием* применяются, когда действия выполняются, пока выполняется некоторое условие или пока условие не будет выполнено. Условием может быть любое выражение, принимающее значение `True` или `False`.

Два основных цикла: `Do While ... Loop` и `Do Until ... Loop`, которые могут быть с предусловием и с постусловием.

Синтаксис цикла с предусловием:

```
Do While | Until Выражение
Операторы
[Exit Do]
Loop        ' Окончание цикла
```

Синтаксис цикла с постусловием:

```
Do
Операторы
[Exit Do]
Loop While | Until Выражение
```

Цикл `Do While` выполняется до тех пор, пока выражение условия имеет значение `True`.

Цикл `Do Until` выполняется до тех пор, пока выражение условия имеет значение `False`.

С помощью оператора `Exit Do` можно цикл прервать, например, для выполнения дополнительного условия.

Встроенные функции. Функция — блок программы, выполняющий действия и возвращающий значение. Синтаксис функции:

```
[Private | Public] Function Имя Функции
          (Список параметров) As Тип
     Тело функции (строки кода)
     Имя Функции= Возвращаемое Значение
End Function
```

где `[Private | Public]` — необязательные ключевые слова, определяющие область видимости функции;

`Function` — ключевое слово, обозначающее функцию;

`Тип` — тип данных возвращаемого значения.

Вызов функции осуществляется по конструкции `Имя Функции (Список параметров)`.

Visual Basic располагает рядом встроенных функций.

По назначению встроенные функции объединены в следующие группы:

1. *Финансово-математические функции*, например `SYD` (вычисляет величину годовой амортизации фондов за определенный период), `DDB` (вычисляет амортизацию фондов в течение заданного интервала времени) и др.

2. *Функции преобразования типов*, например `Val` (Строка) — преобразует строку цифровых символов в число, и много других. Часто преобразование типов выполняется автоматически. Но могут возникать ошибки. Поэтому пользователь сам в процессе

программирования должен определить необходимость преобразования типов данных.

3. Математические функции, например `Cos` (Числовое выражение) — косинус указанного в радианах угла, `Exp` (Числовое выражение) — возвращает число `e` (основание натурального логарифма), возведенное в указанную степень, и др. Предназначены для математических расчетов.

4. *Строковые функции*, например `Asc` (Строковое выражение) — возвращает ASCII-код первого символа в строковом выражении, `Space` (Длина) — возвращает строку пробелов указанной длины.

Кроме того, имеются встроенные функции:

- *даты и времени*;
- *статуса*;
- *для работы с массивами*;
- *для работы с файлами*;
- *прочие функции.*

Типы процедур: `Sub`, `Function`, `Property`.

При запуске процедуры `Sub` (Набор команд) выполняются ее команды. Это самая распространенная процедура.

Создание процедуры `Sub` имеет следующий синтаксис:

```
[{Private | Public}] [Static] Sub имя ([список Аргументов])
[инструкции]
[Exit Sub]
[инструкции]
End Sub
```

Например, для создания процедуры `Sub` сложения двух чисел следует на листе Excel создать элемент управления (командную кнопку), в свойстве `Name` задать имя. Два раза щелкнув по кнопке, выйти в редактор кода, при этом курсор будет находиться в процедуре `Private SubCmd_Click()`, где `Cmd` — имя кнопки, `Click` — событие щелчок. Следует написать необходимый код до строки `End Sub`.

Немедленный выход из процедуры (`Exit Sub`) может возникнуть, например, при выполнении какого-то условия. Не допускаются вложенные процедуры `Sub`.

При запуске процедуры `Function` (Функция) решается определенная задача. Функция обязательно должна возвратить оп-

ределенное значение. Создание Function имеет следующий синтаксис:

```
{[Private | Public]} [Static] Function имя ([список
Аргументов]) [As тип]

[инструкции]
[имя = выражение]          ' строка передачи значения
[Exit Function]
[инструкции]
[имя = выражение]
End Function
```

Процедура Property (Свойство) используется для ссылки на свойство объекта.

Контрольные вопросы

1. Каков алфавит языка Visual Basic?
2. Что такое программный код?
3. Какие классы лексем различают в языке Visual Basic?
4. Какие типы данных используют в языке Visual Basic?
5. Какие способы объявления переменных имеет Visual Basic?
6. Какие применяют обозначения типов переменных?
7. Что такое область видимости?
8. Как пользователь может определить свои типы данных?
9. Какие операции используют в языке Visual Basic?
10. Какие операторы применяют для выполнения условных конструкций?
11. Какие операторы применяют для выполнения циклических конструкций?
12. Какие встроенные функции используют в Visual Basic?

3.6. Объекты, свойства, методы и события MS Visual Basic

Объекты (экземпляры класса) представляют собой сущности, которые могут содержать данные и код. Объекты предоставляют программисту все основные возможности объектно-ориентированного программирования, такие как наследование, инкапсуляция и полиморфизм.

Visual Basic содержит объекты (простые, элементы управления, коллекции и др.), основными из которых являются следующие.

Простые объекты:

- `ClipBoard` — предоставляет доступ к буферу обмена Windows;
- `Font` — содержит информацию, необходимую для определения внешнего вида текста;
- `Form` — окно, содержащее рабочую поверхность проекта;
- `MDIForm` — материнская MDI-форма, которая может содержать в себе дочерние формы (документы);
- `Picture` — позволяет манипулировать графикой в различных форматах;
- `Screen` — позволяет манипулировать положением формы на экране, шрифтами, а также указателем мыши за пределами формы.

Управляющие элементы:

- `CheckBox` — поле флажка, позволяющее задавать и отображать логическое значение;
- `ComboBox` — комбинированное поле, объединяющее текстовое поле и связанное с ним поле списка;
- `Data` — предоставляет доступ к базе данных посредством объекта `Recordset` одного из трех типов;
- `FileListBox` — отображает список файлов текущего каталога и позволяет выбрать один или несколько файлов;
- `Frame` — рамка является контейнером, в который можно поместить другие управляющие элементы;
- `HScrollBar`, `VScrollBar` — полосы прокрутки, позволяющие отображать и вводить числовые значения из заданного диапазона с помощью указателя мыши;
- `Image` — графический образ, предназначенный для отображения на экране точечного рисунка или значка;
- `Label` — поле для отображения текстовой надписи;
- `Line` — отображает линию;
- `ListBox` — поле списка, где можно отображать элементы списка и выбирать их;
- `Menu` — элемент меню;
- `OLE` — OLE-контейнер, способный заключать в себе связываемые и внедряемые объекты;

- OptionButten — позиция переключателя. Два и более элементов этого вида, помещенные в один контейнер, составляют переключатель на соответствующее число позиций;
- PictureBox — отображает на экране загруженные точечные рисунки и позволяет рисовать на своей поверхности с помощью графических методов;
- Shape — фигура, служит для представления на рабочей поверхности объекта геометрических фигур;
- TextBox — отображает и позволяет изменять текст;
- Timer — генерирует события Timer через заданные промежутки времени.

Коллекции:

- DataObjectFiles — коллекция строк с именами файлов, используемых объектом DataQbject;
- Forms — окна форм, включают все загруженные формы приложения.

Свойство — это управляющая структура, с помощью которой происходит доступ к полям объекта. Поле — это переменная класса, изменять значение которой можно с помощью свойств. Свойства определяют внешний вид и поведение объектов. Однотипные объекты могут иметь по ряду свойств одни и те же установки. Такие свойства называются *групповыми* и устанавливаются одновременно для выделенных объектов. Многие свойства являются *общими* (не путать с групповыми) для объектов. Например, свойства Name (Имя), Caption (Надпись), размеры и положение, шрифты, цвет, фокус (активизация) объектов, возможность видимости и т. д. Visual Basic содержит свойства объектов, основными из которых являются следующие:

- AutoRedraw — разрешает или запрещает автоматическое перерисовывание результатов работы графических методов на поверхности формы или элемента PictureBox;
- Autosize — разрешает или запрещает автоматическое изменение размеров элемента в соответствии с размерами его содержимого;
- Backcolor, Forecolor — задают цвета соответственно фона и переднего плана объекта;
- BackStyle — определяет стиль фона объекта UserControl;
- BOF Action, EOF Action — содержит и позволяет задать значение вида автоматического действия, которое будет предпринято при получении свойством BOF или EOF значе-

ния True (т. е. при попытке перейти на позицию до начала или после конца файла);

- Bold — определяет наличие признака начертания "полужирный" для объекта Font;
- BorderColor — задает цвет обрамления;
- BorderStyle — задает стиль обрамления;
- BorderWidth — возвращает и позволяет задать значение, определяет толщину линий обрамления;
- Cancel — содержит и позволяет задать значение, определяющее срабатывание командной кнопки (обычно это кнопка **Отмена**) при нажатии клавиши **Esc**;
- Caption — содержит строковое значение, определяющее надпись на управляющем элементе или объекте;
- Changed — сигнализирует об изменении значения свойства на странице свойств;
- Class — возвращает имя класса, к которому относится внедренный объект;
- Clipboard — позволяет получить доступ к буферу обмена Windows;
- Col, Row — определяют текущую клетку в сетке или таблице данных;
- Cols, Rows — определяют общее количество столбцов и строк в сетке;
- Columns — определяет количество столбцов в списке;
- CurrentX, CurrentY — содержат пару координат, на которых завершилась работа графического метода или последнего метода, выводящего символы;
- Database — содержит ссылку на объект типа Database, связанный с элементом типа Data;
- DatabaseName — возвращает и позволяет задать полное имя файла или каталога, содержащего базу данных, с которой связан или должен быть связан элемент типа Data;
- DataField — определяет поле в наборе записей, которое должен отображать данный элемент;
- DataSource — определяет имя элемента типа Data, через который данный управляющий элемент связан с базой данных;
- Default — разрешает или запрещает срабатывание командной кнопки (обычно **OK**) при нажатии на клавишу **Enter**;

- DefaultType (элемент Data) — возвращает и позволяет задать тип источника данных для элемента типа Data;
- DragMode — возвращает и позволяет задать значение, определяющее режим выполнения операций Drag&Drop (автоматический или ручной);
- DrawMode — возвращает и позволяет задать вид линий для графических элементов и методов;
- DrawStyle — возвращает и позволяет задать стиль линий для графических методов;
- DrawWidth — определяет толщину линий для графических методов объектов;
- Enabled — возвращает и позволяет задать значение, разрешающее или запрещающее объекту реагировать на события, вызванные действиями пользователя;
- FileName — возвращает и позволяет задать полное имя файла;
- FileNumber — номер файла, присвоенный ему при открытии;
- Files — возвращает коллекцию DataObject Files, т. е. список имен файлов, которые используются объектом DataObject;
- FontBold, Fontitalie, FontStrikeThru, FontUnderline — определяют атрибуты и признаки начертания шрифта;
- FontSize — определяет размер выбранного шрифта;
- Height, Width — определят размеры объекта или столбцов сетки;
- Icon — возвращает значок, появляющийся вместо формы при ее минимизации;
- Image — возвращает код доступа к результатам работы графических методов, сохранившимся в памяти;
- Interval — определяет интервал времени (в миллисекундах) между событиями таймера;
- KeyPreview — разрешает или запрещает форме перехватывать и предварительно просматривать нажатия клавиш, прежде чем передать их управляющему элементу;
- Left, Top — определяют расстояния между левой и верхней границами объекта и соответствующими границами контейнера;

- `List` — возвращает и позволяет задать содержимое списка, принадлежащего данному управляющему элементу;
- `Max`, `Min` — определяют соответственно максимальное и минимальное значения свойства `Value` для полос прокрутки;
- `MinButton` — возвращает и позволяет задать значение, определяющее наличие кнопки минимизации окна в строке заголовка окна формы;
- `Name` — определяет имя объекта;
- `Path` — возвращает полное имя текущего каталога;
- `Picture` — возвращает и позволяет задать значок или точечный рисунок;
- `Public` — определяет, может ли данный управляющий элемент использоваться одновременно несколькими приложениями;
- `Readonly` — разрешает или запрещает модификацию базы данных;
- `RecordSource` — определяет имя физической таблицы либо SQL-выражение или объект `QueryDef`, который должен служить источником записей для управляющих элементов, связанных с данным элементом `Data`;
- `ScaleLeft`, `ScaleTop` — возвращают и позволяют задать координаты левого верхнего угла объекта;
- `Screen` — позволяет управлять положением формы на экране и указателем мыши за пределами формы;
- `Shape` — возвращает и позволяет задать значение, определяющее вид геометрической фигуры;
- `Size (Font)` — возвращает и позволяет задать значение, определяющее размер шрифта;
- `Style` — определяет стиль изображения объекта на экране и особенности его поведения;
- `Text` — возвращает и позволяет задать текст, содержащийся в текстовом или комбинированном поле;
- `Value` — возвращает и позволяет задать значение, соответствующее состоянию поля флажка или позиции переключателя, положению ползунка на полосе прокрутки, содержимому поля в текущей записи или состоянию командной кнопки;

- `Visible` — разрешает или запрещает отображение объекта на экране;
- `X1`, `Y1`, `X2`, `Y2` — задают координаты начала и конца линии для управляющего элемента типа `Line`.

События — сообщения специального вида, создаваемые средой Windows. Отправителем сообщения может быть клавиатура, мышь и др., получателем — окно приложения и управляющий элемент в этом окне, который обладает фокусом. Для того чтобы событие происходило, в коде программы должны быть процедуры его обработки, т. е. обработчики событий.

Основными событиями являются следующие:

- `Activate` — происходит, когда объект становится текущим окном;
- `Deactivate` — происходит, когда объект перестает быть текущим окном;
- `Change` — сигнализирует об изменении содержимого управляющего элемента;
- `Click` — происходит по щелчку мыши, когда указатель мыши находится на управляющем элементе;
- `DblClick` — происходит по двойному щелчку мыши;
- `DragDrop` — происходит при завершении операции перетаскивания;
- `DragOver` — генерируется в процессе операции перетаскивания;
- `DropDown` — происходит перед развертыванием списка в комбинированном поле;
- `EnterFocus` — происходит, когда объект получает фокус клавиатуры;
- `Error` — происходит при возникновении ошибки доступа к данным, для которой не предусмотрена процедура обработки;
- `ExitFocus` — происходит, когда объект теряет фокус клавиатуры;
- `GotFocus` — происходит, когда объект получает фокус клавиатуры вследствие действий пользователя или благодаря вызову метода `SetFocus`;
- `Initialize` — происходит при создании экземпляра формы, MDI-формы, страницы свойств, определяемого пользователем управляющего элемента или класса;

- `KeyDown` — происходит при нажатии клавиши, если объект обладает фокусом клавиатуры;
- `KeyPress` — происходит при нажатии и последующем отпускании клавиши, соответствующей одному из ASCII-символов, если объект обладает фокусом клавиатуры;
- `KeyUp` — происходит при отпускании клавиши, если объект обладает фокусом клавиатуры;
- `Load` — происходит при загрузке формы;
- `LostFocus` — происходит, когда объект теряет фокус клавиатуры из-за действий пользователя или благодаря вызову метода `SetFocus` другого элемента;
- `MouseDown` — происходит при нажатии кнопки мыши;
- `MouseMove` — происходит при перемещении мыши;
- `MousUp` — происходит при отпускании кнопки мыши;
- `Scroll` — происходит при перемещении ползунка на полосе прокрутки или элементе, содержащем полосу прокрутки;
- `Show` — происходит при изменении значения свойства `Visible` пользовательского управляющего элемента или документа на `True`;
- `Terminate` — последнее событие жизненного цикла формы, страницы свойств, MDI-формы, пользовательского управляющего элемента или класса;
- `Timer` — происходит через интервал времени, заданный значением свойства `Interval` управляющего элемента `Timer`;
- `Unload` — происходит при выгрузке формы и удалении ее с экрана;
- `Updated` — происходит после изменения данных, содержащихся в объекте.

Метод — это процедура или функция класса. Основными являются следующие методы:

- `Arrange` — упорядочивает дочерние окна или их значки в окне материнской MDI-формы;
- `Circle` — предназначен для рисования окружностей, эллипсов, дуг и секторов на рабочей поверхности объекта;
- `Clear` (`Clipboard`, `Combobox`, `ListBox`) — удаляет (очищает) содержимое буфера обмена, комбинированного поля или поля списка;

- Close (OLE-контейнер) — закрывает объект, содержащийся в OLE-контейнере, и прекращает связь с создавшим его приложением;
- Cls — удаляет следы текстового и графического вывода с рабочей поверхности формы или управляющего элемента Picture;
- Copy — копирует объекты из OLE-контейнера в буфер обмена Windows;
- Drag — инициирует, завершает или прерывает операцию перетаскивания объектов;
- Files — возвращает список имен файлов, используя при этом формат vbCFiles;
- GetData — получает графические данные из буфера обмена Windows или получает данные типа Variant из объекта типа DataObject;
- GetText — получает текстовые данные из буфера обмена Windows;
- Line — рисует линии и прямоугольники на рабочей поверхности формы или управляющего элемента типа PictureBox;
- Move — перемещает формы, MDI-формы и управляющие элементы;
- PaintPicture — загружает точечный рисунок из файла и выводит его на рабочую поверхность формы, картинки или объекта Printer;
- Paste — выполняет операцию вставки объекта в OLE-контейнер из буфера обмена Windows;
- PSet — рисует точку заданного цвета в заданных координатах на поверхности объекта;
- Refresh — принудительно перерисовывает форму или управляющий элемент в соответствии с его текущим содержимым;
- Render — рисует заданную часть рисунка на поверхности заданного объекта;
- Scale — задает систему координат для формы, управляющего элемента PictureBox или объекта Printer;
- ScaleX, ScaleY — преобразуют значения координат из одной системы в другую;
- SellectAll — выделяет все элементы формы;

- `SetData` — помещает графические данные в буфер обмена или в объект `DataObject`;
- `SetFocus` — помещает фокус клавиатуры на данный объект;
- `SetText` — помещает в буфер обмена текстовые данные;
- `Show` — отображает на экране форму или MDI-форму;
- `Size` — изменяет размеры управляющего элемента, определяемого пользователем;
- `Zorder` — определяет способ появления объекта в его контейнере: впереди или позади других объектов.

Контрольные вопросы

1. Что такое объекты Visual Basic?
2. Что определяют свойства?
3. Что задают методы?
4. Что такое события в Visual Basic?
5. Приведите примеры основных объектов, свойств, методов и событий Visual Basic.

3.7. Программирование в среде MS Visual Basic

3.7.1. Разработка первого проекта

Запуск системы осуществляется командами **Пуск** → **Программы** → **MS Visual Basic**.

Далее следует создать новый или открыть существующий проект. Новый проект открывается в среде разработки вместе с другими окнами и инструментами (см. рис. 3.10).

Если работа будет проводиться во встроенном в ОС Windows редакторе Visual Basic, то сначала следует создать файл, щелкнув правой кнопкой мыши. В открывшемся контекстном меню выбрать команды **Создать** → **Документ MS Word**, сохранить документ и переименовать его. Далее войти в интегрированную среду разработки Visual Basic можно с помощью команд меню процессора Word **Сервис** → **Макрос** → **Редактор Visual Basic**.

Для приобретения навыков работы по созданию простейшего проекта в среде Visual Basic можно выполнить, например, решение задачи сложения двух чисел.

Программу назовем «Сложение». Два числа, вводимых с клавиатуры, следует занести в два поля с именами **Число 1** и **Число 2**, сложить их. Сумма должна отобразиться в поле **Результат** после нажатия кнопки **Сложить**. Работа заканчивается после нажатия кнопки **Выход**.

Пользовательский интерфейс проекта «Сложение» будет иметь две командные кнопки, три поля для отображения чисел с именами **Число 1**, **Число 2**, **Результат** и надпись «Сложение» (рис. 3.11).

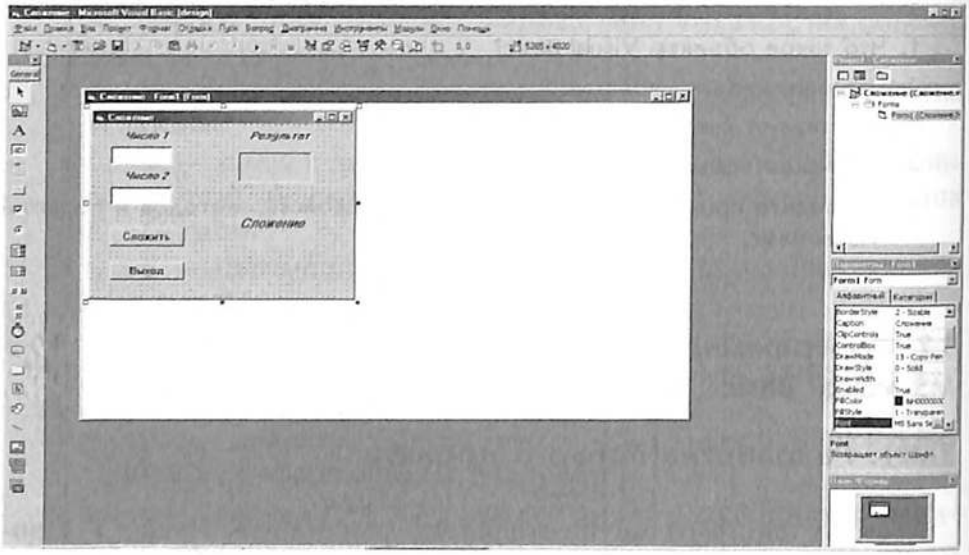


Рис. 3.11. Интерфейс проекта «Сложение»

Щелкнув мышью по кнопке **Command Button** на панели инструментов, следует переместить указатель мыши на форму в левый нижний угол. Вычертить указателем прямоугольник, который будет привязан к сетке формы. Аналогично следует создать вторую кнопку, разместив ее под первой. Кнопки будут иметь названия соответственно Command1 и Command2. Шаг сетки можно изменить, выполнив последовательно команды **Tools (Настройки)** → **Options (Параметры)** → **General (Общие)**.

Щелкнув мышью по кнопке **TextBox** (**ab** — управление текстовым окном) на панели инструментов, следует переместить указатель мыши на форму в левый верхний угол. Вычертить указателем прямоугольный объект. По умолчанию объект получит имя Text1. Аналогично следует под ним создать поле Text2. Это поля для ввода чисел.

Добавить в форму этикетку для отображения результата нужно, щелкнув мышью по кнопке **Label** (**A** — управление надписью), и переместить указатель мыши на форму в левый верхний угол. Вычертить указателем прямоугольник. По умолчанию объект получит имя Label1. Аналогично следует создать этикетки для надписей над окнами ввода чисел, результата и названия программы (Label2, Label3, Label4, Label5).

Установка свойств объектов. Для изменения названия формы Form1 на нужное следует щелкнуть мышью по форме, и в окне **Properties** появятся свойства формы. Дважды щелкнув по свойству **Caption** (**Название**) в колонке **Alphabetic**, в колонке **Categorized** вместо Form1 ввести «Сложение».

Далее, щелкая мышью по объектам, следует ввести для командных кнопок в свойстве **Caption** вместо Command1 и Command2 соответственно «Сложить» и «Выход».

В окне свойств можно установить для кнопок также цвет фона, размер шрифта, ширину кнопки и другие свойства.

Поля ввода чисел выполняют одинаковые функции, поэтому их следует объединить в группу. Щелкнув мышью по полю Text1 и удерживая нажатой клавишу **Shift**, щелкните по полю Text2. Оба текстовых поля будут выделены, а устанавливаемые значения свойств будут общими. Далее следует назначить свойству **Text** (**Текст**) для текстовых полей значение **Empty** (**Пусто**), т. е. удалить текущую установку и оставить свойство пустым. После этого с клавиатуры можно вводить информацию.

Чтобы установить свойство для поля отображения результата, нужно, выделив это поле, затем выровнять (свойство **Alignment** (**Выравнивание**)), например, по центру (**Center**). Изменить стиль рамки можно с помощью свойства **Border Style** (**Стиль рамки**), например добавить тонкую рамку вокруг поля, выбрав значение **Fixed Single**. Шрифт в окне результата следует установить, дважды щелкнув по свойству **Font** (**Шрифт**). В появившемся диалоговом окне выбрать шрифт (например, Times new Roman), начертание (например, полужирный), кегль (например, 14) и нажать

кнопку **OK**. Поле вывода результата должно быть пустым, название Label1 в свойстве **Caption** следует удалить.

Для полей ввода чисел и названия результата (Label2, Label3, Label4) следует ввести общие свойства выравнивания, стиля рамки и шрифт, выделив эти объекты как группу (щелкая по ним и удерживая при этом нажатой клавишу **Shift**). Для каждого из этих объектов необходимо установить значение свойства **Caption** соответственно «Результат», «Число 1», «Число 2».

Измените название надписи Label5 на «Сложение» и установите для него свойства выравнивания, стиля рамки и шрифт.

Для выбора цвета следует дважды щелкнуть по свойству **ForeColor** (**Основной цвет**), в поле списка появятся две вкладки — **System** (**Система**) и **Palette** (**Палитра**). Первая показывает цвет пользовательского интерфейса, вторая содержит все доступные цвета.

Создание программного кода. В разрабатываемой программе недостает только кода, который будет вычислять сумму вводимых с клавиатуры чисел и отображать результат вычислений. Программа управляется кнопками **Сложить** и **Выход**, поэтому программный код будет связан с информацией, поступающей от этих кнопок.

Для работы с программным кодом командной кнопки нужно дважды по ней щелкнуть мышью. Появляется окно **Code**, в котором тело процедуры заключено между операторами, указывающими на начало и конец подпрограммы. Операторы процедуры будут выполняться, если активизировать элемент интерфейса, ассоциированный с процедурой. В данном случае событием является щелчок мыши (`Click`), но могут быть и события другого типа. (Их можно увидеть в раскрывающемся списке, шелкнув по стрелке правее надписи Click.)

Разработаем программный код для кнопок **Сложить** и **Выход**:

1. Открыть список объектов (**General**) в окне **Code**. Все объекты интерфейса «Сложение» появятся в списке.

2. Щелкнуть по строке Command в списке. Появится пустая процедура, связанная с кнопкой Command 1. (Несмотря на то что название кнопки было изменено на «Сложить», в программе ее имя осталось прежним.) Для кнопки Command1 задано только одно событие — щелчок мышью.

3. Набрать программные строки тела процедуры между утверждениями `Private Sub` и `End Sub`, как показано на

рис. 3.12. Функция `Val` преобразует текстовый аргумент в числовое значение (по умолчанию данные, введенные в текстовое поле, представляются текстовой строкой).

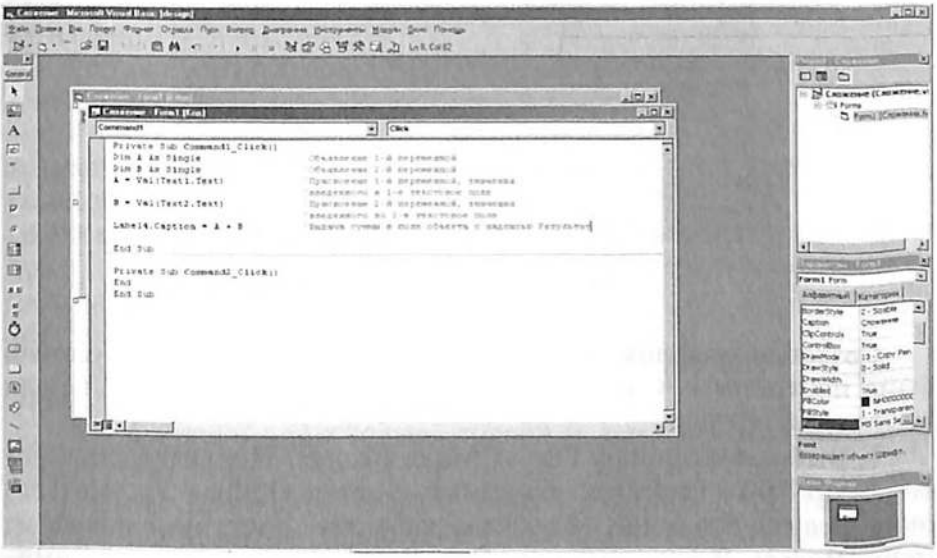


Рис. 3.12. Редактор кода

4. Выбрать строку Command2 в списке объектов (**General**) в окне **Code**. Набрать программную строку процедуры, связанной с кнопкой Command2 — это один оператор `End`. Он используется для останова программы и удаления ее с экрана.

5. Сохранить программу.

Запуск программного кода следует осуществить из меню **Run** или щелчком по кнопке **Start** на панели инструментов. На экране появится разработанный интерфейс (рис. 3.13). В поле ввода числа 1 ввести произвольное число (целое или с дробной частью). В поле ввода числа 2 также ввести произвольное число. Щелкнуть по кнопке **Сложить**. В поле результата появится сумма двух введенных чисел. Для продолжения работы программы, т. е. для сложения других чисел, надо щелкнуть мышью по полю ввода числа 1 и ввести его.

Далее выполнять действия ввода в поле числа 2 и т. д. Работа программы в режиме диалога будет продолжаться до тех пор, пока не произведен щелчок мышью по кнопке **Выход**.

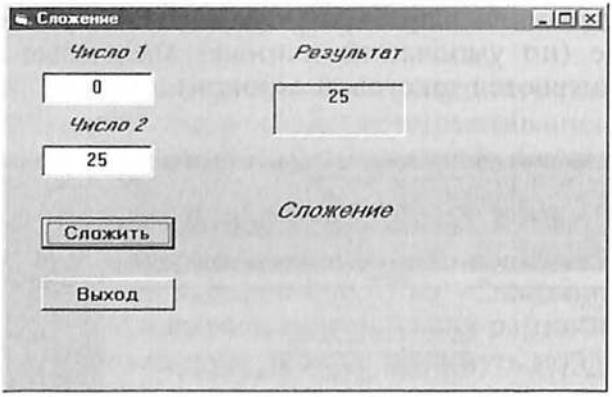


Рис. 3.13. Рабочий проект

Создание выполняемого файла. Чтобы создать исполняемый файл программы «Сложение» с расширением .exe и ярлыка для ее запуска, следует:

1. Выбрать команды **File** → **Make Project**. Появится диалоговое окно **Make Project**. С помощью кнопки **Options** открывается окно **Project Properties** (**Свойства проекта**), доступное также из меню **Project**. В окне закладка **Compile** используется для оптимизации процесса компиляции программы.

2. Щелкнуть по кнопке **OK** для подтверждения имени файла. VB создаст выполняемый файл в указанном месте.

3. В контекстном меню выбрать команды **Создать** → **Ярлык**. В диалоговом окне щелкнуть по кнопке **Обзор** и выбрать файл Сложение.exe.

4. Щелкнуть по кнопкам **Открыть**, **Далее**, **Готово**. На рабочем столе будет создан ярлык для программы Сложение.exe. Теперь программа будет работать как обычное приложение ОС Windows.

Контрольные вопросы

1. Какими командами осуществляется запуск Visual Basic?
2. Как создать простейший проект в среде программирования VB?
3. Что такое пользовательский интерфейс VB и как его создать?
4. Как установить свойства объектов?
5. Как создать программный код в среде программирования VB?
6. Как создать исполняемый файл?

3.7.2. Создание проектов

Проект состоит из совокупности файлов, которые необходимы для получения исполняемого файла приложения. Файл проекта (.vbp) содержит список имен частей проекта. Проект состоит из форм и модулей.

Форма — это основа графического интерфейса приложения, в ней размещаются все элементы управления.

Модуль — это файл, содержащий программный код, программные инструкции, в которых объявляются переменные, константы, размещаются пользовательские функции и подпрограммы. Кроме модуля .bas имеются следующие: модуль классов — .cls; документ пользователя — .dob; страница свойств — .pag; управляющий элемент пользователя — .ctl; OCX-элемент (ActiveX-элемент) — .ocx.

Группа проектов может быть сохранена в файле с расширением .vbg.

Все файлы — текстовые, содержат предложения на языке Visual Basic.

В диалоговом окне **Project Properties** можно установить конфигурацию проекта. Диалоговое окно можно вызвать командами **Tools → Project Properties**. На вкладке **General (Общие)** этого окна указывают имя и вид проекта. Другие вкладки: **Make (Сборка)**, **Component (Компонент)**, **Compile (Компиляция)**.

Синтаксические и логические ошибки при выполнении программы. Основные средства отладки можно получить через меню **Debug (Отладить)** и панель инструментов, вызвать которую можно командами **View (Вид) → Toolbars (Панели инструментов) → Debug** (рис. 3.14). Вкладки панели следующие:

- **Start** (запуск программы);
- **Pause** (приостановка программы);
- **End** (останов программы);
- **Toggle Breakpoint** (включить/выключить контрольную точку);
- **Step Into** (пошаговая обработка с заходом в процедуры);
- **Step Over** (пошаговая обработка без захода в процедуры);
- **Step Out** (выход из процедуры);
- **Locals Window** (включить/выключить локальное окно);
- **Immediate Window** (окно непосредственного выполнения);
- **Watch Window** (включить/выключить окно наблюдаемых переменных);

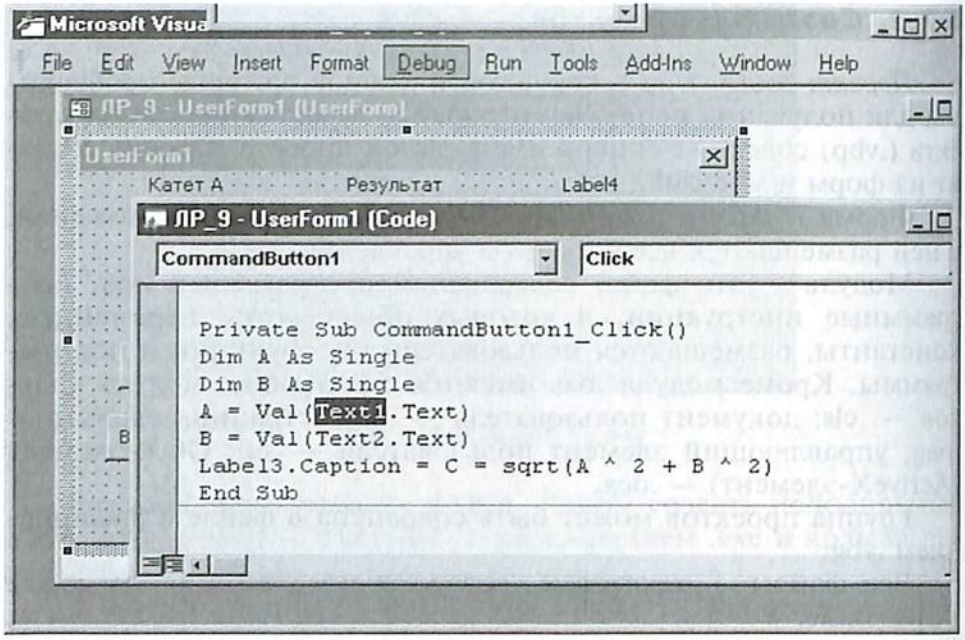


Рис. 3.14. Отладка программы

- **Quick Watch** (быстрый просмотр значения переменной);
- **Call Stack** (просмотр списка вызова процедур).

Для просмотра списка возможного выявления ошибок следует открыть меню **Tools** (**Сервис**) и выбрать пункт **Options** (**Параметры**). На экране появится окно, где следует установить флажки. Чтобы работа не прерывалась сообщениями о каждой ошибке, нужно снять флажок **Auto Syntax Check** (**Автоматическая проверка синтаксиса**).

Среда разработки Visual Basic обнаруживает ошибки типов или неполные программные блоки. Visual Basic допускает выполнение программы с необъявленными переменными. Оператор Option Explicit (раздел **General** или модуль) определяет необъявленные переменные.

Если компилятор обнаружит ошибки, то для их исправления и продолжения работы достаточно:

- изменить подсвеченную строку и нажать клавишу **F5** или выбрать команды **Run** → **Continue**;
- изменить программу и запустить ее снова командами **Run** → **Restart** (или нажать сочетание клавиш **Shift+F5**);

- запустить программу с другого места, щелкнув мышью по другой строке и выбрав в меню **Debug** команду **Set Next Statement**, и нажать **F1**.

Завершить программу можно командой **End** в меню **Run**.

Точка останова (*контрольная точка*) — строка, где прерывается выполнение программы (выделяется красным цветом). Запущенная программа прервется на этой строке, строка будет выделена желтым цветом и стрелкой.

Способы создания точек останова:

- щелчком мыши на нужной строке и нажатием клавиши **F9**;
- выбрав в меню **Debug** команду **Toggle Breakpoint** (**Установить точку останова**);
- на панели **Debug** щелкнуть по кнопке **Toggle Breakpoint**.

Отмена точек останова выполняется с помощью команд **Debug → Clear All Breakpoints** (**Убрать все точки останова**).

Порядок наблюдения за поведением требуемой переменной:

- установить точку останова в строке, где есть наблюдаемая переменная;
- запустить программу (клавиша **F5** или команда **Start**);
- задержать указатель мыши над переменной, появится окно с ее текущим значением.

Порядок наблюдения нескольких переменных с помощью окна **Watches** (**Наблюдаемые переменные**):

- установить точки останова в строках кода, где есть наблюдаемые переменные;
- запустить программу;
- в точке останова подсветить переменную, добавляемую к списку просматриваемых значений;
- щелчком правой кнопки мыши вызвать контекстное меню и в нем выбрать команду **Add Watch** (**Добавить наблюдаемую переменную**);
- в диалоговом окне **Add Watch** выбрать имя переменной (**Expression**), область видимости переменной (**Context**), тип наблюдения (**Watch Type**) и др., нажать кнопку **OK**;
- для отображения окна **Watches** следует выбрать в меню **View** команду **Watch Window** (**Окно наблюдения**).

Пошаговую (построчную) проверку выполнения программы можно провести в одном из двух режимов:

- **Step Into** (пошаговая обработка с заходом в процедуры) — выполняются все строки программы, включая строки процедур;

- **Step Over** (пошаговая обработка без захода в процедуры) — вызываемые процедуры выполняются как один оператор, а затем — очередная строка программы.

Используя панель **Debug**, при отладке следует выполнить следующее:

- открыть отлаживаемый проект;
- вызвать панель **Debug**;
- панель **Debug** перетащить на форму (для удобства — вниз);
- запустить проект, нажав кнопку **Start**;
- если выдаваемые результаты неверны, следует щелкнуть по кнопке **Break (Останов)** — программа остановится, а в появившемся окне **Code** будет показан текст выполнения программы;
- щелчком по кнопке **Step Into** можно открыть инструкцию, чтобы выполнить следующий оператор;
- для запуска приложения нужно в форме нажать кнопку, например Command1, в окне **Code** отобразится процедура обработки прерывания `Command1_Click`, первый оператор которой будет выделен желтым цветом;
- при щелчке по кнопке **Step Into** будет выполнен оператор `Sub`, а следующий оператор будет выделен; если в этом операторе есть проверяемая переменная, то ее нужно поместить в окно **Watches**;
- выделить проверяемую переменную и щелкнуть по кнопке **Quick Watch (Быстрый просмотр переменных)**. Появится диалоговое окно, в котором будут содержаться имя переменной, ее значение, тип, назначение;
- при щелчке в диалоговом окне **Quick Watch** в нижней части экрана появится окно **Watches**, где будет отображена выбранная переменная. В это окно можно поместить другие переменные;
- далее щелкнуть по кнопке **Step Into** для выполнения следующего оператора и т. д. Таким образом, наблюдая значения переменных в окне **Watches**, можно определить логику программы и обнаружить ошибки.

При пошаговой отладке можно воспользоваться также оператором `Stop`, поместив его в нужное место. Выполнение программы по достижении `Stop` будет приостановлено. В режиме останова можно использовать окно **Code** и панель инструментов **Debug**.

Ошибки выполнения программы. Использование обработчиков ошибок. Иногда даже отлаженная программа не может быть выполнена. Такой сбой представляет собой событие, после которого Visual Basic не может управлять ходом программы. Ошибка выполнения возникает тогда, когда Visual Basic выполняет оператор, который не может быть правильно завершен, т. е. компилятор не получил указаний, если что-то не в порядке. Чтобы не допустить таких ситуаций, нужно создать специальные процедуры — обработчики ошибок. Обработчик ошибок сообщает программе, как ей продолжить работу в случае появления ошибки одного из операторов. Обработчик ошибок помещается в те же процедуры обработки прерываний, что и операторы, которые могут вызвать сбой. На такую ситуацию обработчик ошибок реагирует объектом для обработки ошибок с именем `Err`, который имеет свойство `Number` (Число). Ошибка определяется, и это позволяет программе принять необходимые действия. Чаще всего на правильную работу программы влияют внешние события, которые воспринимаются обработчиком ошибок. Это могут быть проблемы с сетью; с гибким диском; с принтером; ошибки переполнения; ошибки использования памяти; с буфером обмена; логические ошибки.

Оператор `On Error` обнаруживает ошибки выполнения.

Синтаксис оператора: `On Error Go To метка`, где `метка` — имя обработчика ошибки.

Оператор — обработчик ошибок надо поместить в процедуру перед сомнительным оператором, он также укажет компилятору место перехода в программе в случае возникновения ошибки. В конце процедуры размещают обработчики ошибок, состоящие из двух частей:

- свойства `Err. Number` в структуре с условием (`If...Then` или `Select Case`). Затем отображается сообщение или устанавливается свойство на основе типа ошибки;
- оператора `Resume`, возвращающего управление в программу. Оператор может быть использован в виде `Resume`, `Resume Next` и `Resume метка`. `Resume` — управление возвращается оператору, вызвавшему ошибку (условия для возникновения ошибки нет). `Resume Next` — возвращает управление оператору, следующему за оператором, вызвавшим ошибку. `Resume метка` — переход на оператор с указанной меткой.

Пример выполнения работы «Программирование в среде разработки MS Visual Basic»

1. **Цель работы.**

Цель работы — получить практические навыки по разработке программного кода и отладке на языке Visual Basic.

2. **Исходные данные.**

Задача: Даны длины двух катетов (A и B) прямоугольного треугольника.

Определить периметр (P) этого треугольника ($P = A + B + C$, где $C = \sqrt{A^2 + B^2}$).

3. **Перечень применяемых приборов и материалов.**

3.1. ПЭВМ: Intel Celeron (33 МГц, 32 Мб).

3.2. Программное обеспечение: ОС Windows, система программирования MS Visual Basic.

4. **Ход и результаты работы.**

4.1. Открываем новый проект для решения предложенной задачи.

4.2. Создаем интерфейс (рис. 3.15).

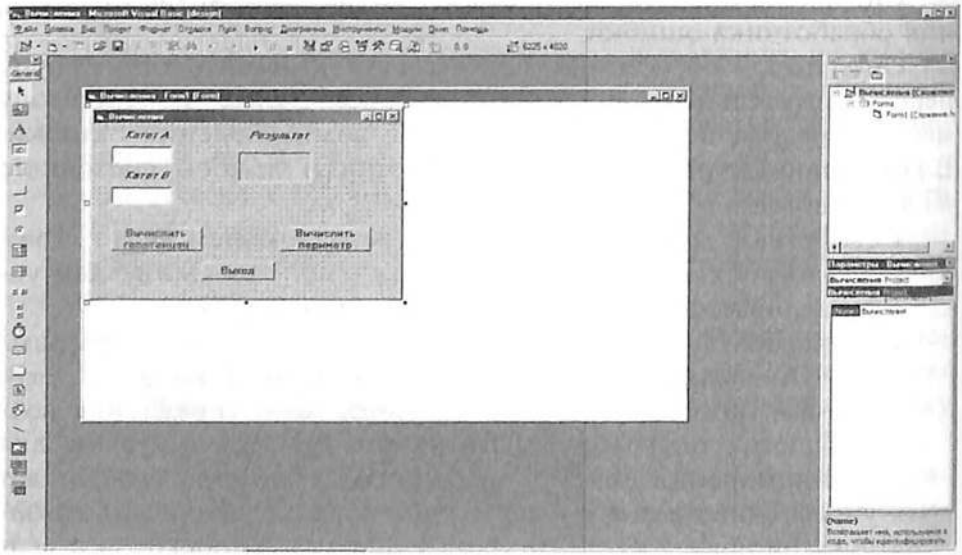


Рис. 3.15. Интерфейс программы для решения задачи

4.3. Составляем программу решения задачи (рис. 3.16).

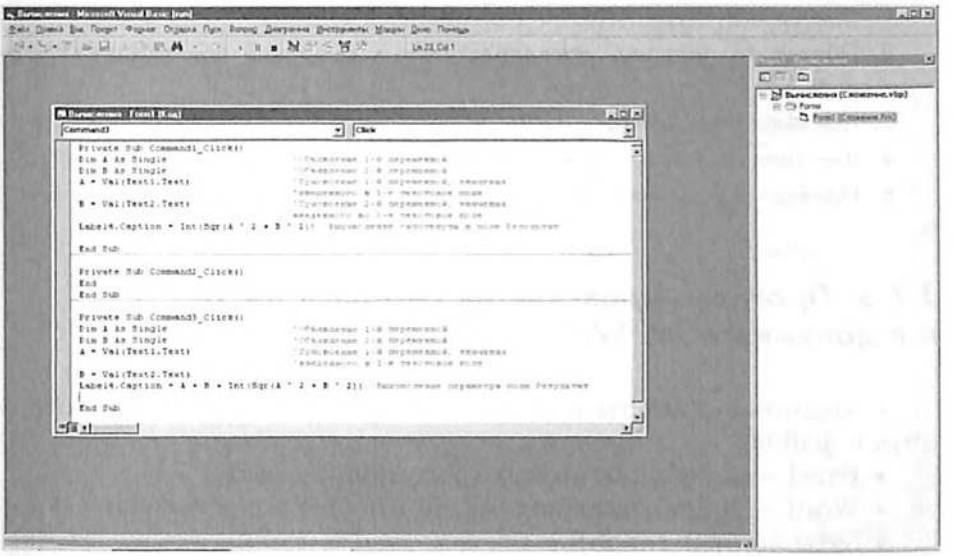


Рис. 3.16. Программный код для решения задачи

4.3. Запускаем программу на выполнение с помощью системной команды **Run** и выполняем отладку программы.

4.4. Сохраняем проект.

Окончательный вид проекта для решения задачи представлен на рис. 3.17.

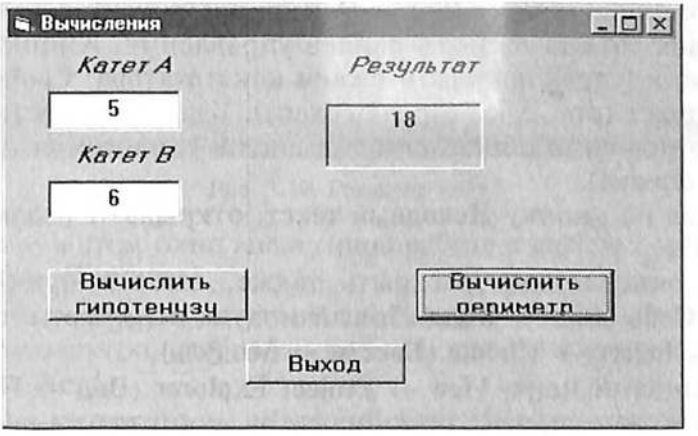


Рис. 3.17. Выполняемый проект для решения задачи

Контрольные вопросы

1. Какова структура проекта Visual Basic?
2. Приведите перечень последовательных операций при создании проекта.
3. Как выполнить отладку проекта?
4. Как создать точки останова?
5. Приведите пример создания простого проекта.

3.7.3. Программирование на рабочем листе Excel и в документе MS Word

Созданные проекты в приложениях MS Office можно сохранить в файлах:

- Excel — в рабочих книгах (расширение .xls);
- Word — в шаблонах документа (.dot) и в документах (.doc);
- базы данных (.mdb);
- PowerPoint — в презентациях (.ppt).

При решении задач в Excel программирование является вспомогательным средством, так как нет необходимости организации ввода и вывода данных, создания вычислительных процедур, оформления интерфейса, построения графиков и т. д. VBA полезно применять для создания пользовательских функций.

Электронные таблицы запускают, используя команды **Пуск → Программы → MS Excel**.

С помощью команд **Вид → Панели инструментов → Элементы управления** можно вызвать панель управления. Кнопки панели управления (слева направо): **Режим конструктора**, **Свойства**, **Исходный текст** (рис. 3.18, стрелка слева). Кнопка **Свойства** открывает окно свойств объектов и элементов управления (рис. 3.18, стрелка справа).

Нажав на кнопку **Исходный текст**, открывают редактор кода (рис. 3.19).

Это окно можно раскрыть также, выбрав пункты меню **View → Code (Вид → Код)**. Новый модуль получают с помощью команды **Insert → Module (Ввести → Модуль)**.

Командами меню **View → Project Explorer (Вид → Проводник проекта)** можно вызвать окно проектов, в котором в виде дерева отображаются названия открытых книг и стандартных шабло-

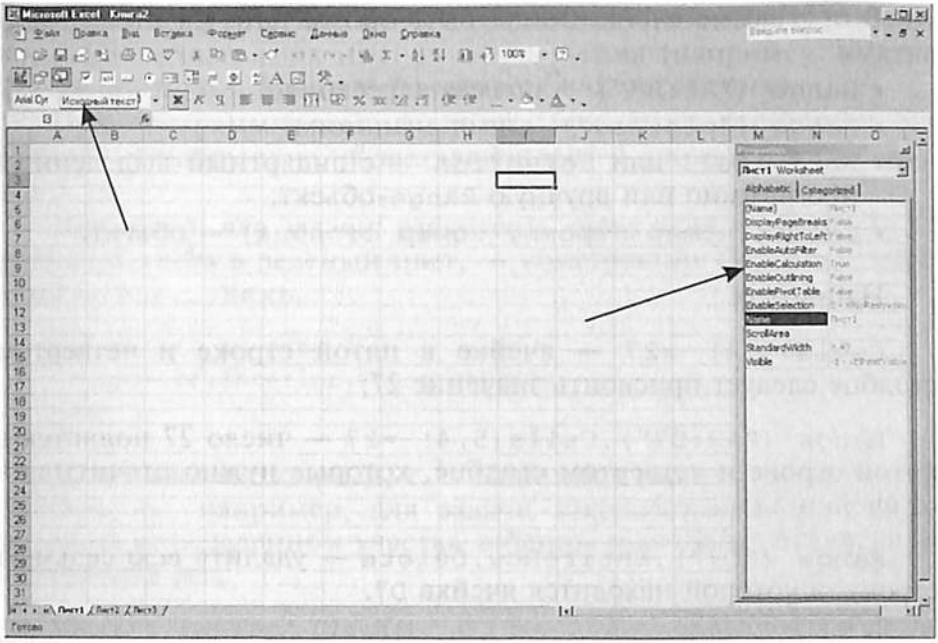


Рис. 3.18. Панель управления и панель свойств на рабочем листе Excel

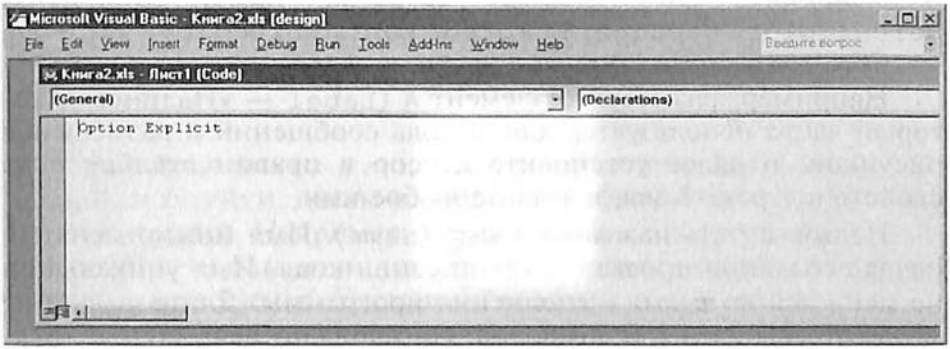


Рис. 3.19. Редактор кода

нов. Работа в этом окне аналогична работе с деревом каталогов в Проводнике Windows.

В VBA в качестве источника данных могут быть использованы ячейки электронной таблицы Excel или фрагменты текста документа в Word.

Range — важнейший объект. В Excel **Range** — диапазон ячеек, в Word **Range** — фрагмент текста.

В программе ячейки могут быть обозначены разными вариантами:

- `Range ("A1:B3")` — прямая адресация;
- `Range ("ploshad")` — имя диапазона;
- `ActiveCell` или `Selection` — стандартный выделенный программно или вручную `Range`-объект;
- `Cells (номер строки, номер столбца)` — объект.

Например:

`Cells(5,4) =27` — ячейке в пятой строке и четвертом столбце следует присвоить значение 27;

`Range ("A2:D7").Cells(5,4) =27` — число 27 появится в пятой строке и четвертом столбце, которые нужно отсчитывать от ячейки `A2`;

`Range ("D7").EntireRow.Delete` — удалить всю седьмую строку, в которой находится ячейка `D7`.

При выполнении практического задания следует по условию задачи на рабочем листе Excel расположить элемент управления и установить его основные свойства: `Name`, `BackColor`, `BorderColor`, `Caption`, `Enabled`, `Font`, `ForeColor`, `Picture`, `Value`, `Visible`, `Height`, `Width`, `Left`, `Top`.

Например, разместите элемент `A` (`Label` — «Надпись»), который часто используется для вывода сообщений и размещения рисунков, и далее установите курсор в правом столбце окна свойств в строке **Name** и задайте любое имя.

Нельзя путать название и имя (`Name`). Имя можно менять в начале создания проекта, до написания кода. Имя уникально и не меняется во время выполнения программы. Длина имени не может превышать 256 символов. Оно должно начинаться с буквы и может состоять из букв, цифр, символов (кроме точек, пробелов, %, &, !, @, $). Лучше применять латиницу. Регистр в именах не имеет значения. Желательно, чтобы имя было информативно, т. е. несло информацию о назначении и типе объекта.

Установив курсор в строке **Caption** (Сообщение), можно набрать любой текст. Убрав курсор, можно этот текст увидеть. Свойство **Caption** отображает сообщение (текст), которое будет размещено на объекте. Оно может меняться во время выполнения программы.

Для быстрого объявления переменных можно использовать присоединенные к их именам обозначения (например, A%, B&, C!, D#, E$ и др.). Массив объявляется как переменная.

При программировании следует применять структуры: линейные, циклические, разветвляющиеся и вложенные (см. раздел 3.5).

Например, для задачи: раскрасить через одну ячейки C1:C12 рабочего листа в зеленый цвет, — конструкция будет циклической — `For...Next`.

```
For K=1 To 12 Step 2
Cells(K,2).Interior.Color=VbGreen.
Next K
```

Если цикл вложенный, т. е. один цикл располагается внутри другого, то, например, для задачи закрашивания нескольких ячеек на определенном участке таблицы программа будет иметь следующий вид:

```
Private Sub Cmd4_Click()
Range("A1:D15").Clear
     For K=1 To 12 Step 2
     For J=1 To 15 Step 3
Cells(K,J).Interior.Color=VbGreen.
Next K
End Sub
```

Конструкцию `For Each...Next` применим для однотипных элементов (ячейки, листы рабочей книги), объединенных в группы (коллекции). Например, для задачи закрашивания ячеек диапазона B2:D10 в цвет, индекс которого равен номеру строки, где располагается ячейка. Элемент обозначим переменной X.

```
Private Sub CommandButton1_Click()
Range("A1:E15").Clear
For Each X In Range ("B2:D10")
X.Interior.ColorIndex = X.Row 'X.Row - номер строки,
  в которой находится ячейка
Next X
End Sub
```

Циклическую конструкцию `Do While ...Loop` применим, например, для задачи закрашивания ячеек во втором столбце,

начиная с первой строки, пока не встретится ячейка с данными.

```
Private Sub Cmd5_Click()          ' начало
Range("A1:E15").ClearFormats      ' очистить указанные ячейки
от форматов и окраски
X=""                              ' переменная пуста
K=1                               ' счетчик
Do While X=""                     ' делай, пока выполняется
                                  ' условие, что X — пуста
K=K+1                             ' если X— пустая строка, то
                    ' начинается закрашивание со 2-й строки
Cells(K,2).Interior.ColorIndex=K
'закрашивается ячейка в K-строке, цвет фона определяем
'через свойство ColorIndex (номер цвета)
X=Cells(D,2)                      ' считывание в X
                                  ' содержимого ячейки
Loop
End Sub
```

Если вместо `While` будет `Until`, то цикл будет выполняться, пока не выполняется условие.

Конструкция `With...End With` имеет следующий синтаксис:

```
With <объект>
[инструкции]
End With
```

Используется, когда нужно изменить сразу несколько свойств объекта.

Например, `With Cells (3,4)`.

Для создания пользовательских функций в Excel следует использовать рабочую книгу с записанными в ней макросами. Это может быть книга со стандартным именем Personal.xls. Если такая книга отсутствует, то следует в чистой книге записать простенький макрос и сохранить ее под названием Personal.xls.

Например, записать макрос можно следующим образом.

В ячейку B3 ввести число 5, а в ячейку C3 — число 4. В ячейку D3 ввести формулу =B3*C3, тогда результат в ячейке D3 будет равен 20. Для записи этих действий в виде макроса следует очистить содержимое ячеек, а курсор установить в любую ячейку, кроме очищенных B3, C3, D3. Выбрать команды меню

Сервис → Макрос → Начать запись. Откроется окно, в котором следует сохранить Макрос1 в «Этой книге». Появится окошко **Остановка записи.** Все дальнейшие действия будут записаны.

Далее, поставим задачу: вычислить гипотенузу в прямоугольном треугольнике. Работать следует в книге Personal.xls.

По теореме Пифагора гипотенуза $C = \sqrt{A^2 + B^2}$, где A и B — катеты.

В книге Personal.xls два раза щелкнув по Module1, выйти в окно кода для этого модуля. При выборе команды меню **Insert → Procedure (Ввести → Процедура)** откроется окно **Add Procedure (Добавить процедуру)**, в котором следует дать имя процедуре, например Gipotenuza. Тип процедуры выбираем **Function**. В окне кода между служебными словами следует набрать необходимый код для создания процедуры, которая в окончательном варианте должна выглядеть следующим образом:

```
Public Function Gipotenuza(A,B)
Gipotenuza = (A*A+B*B) ^ 0.5
End Function.
```

Перейдя на рабочий лист, в любые две ячейки запишем два числа — длины катетов. Для третьей ячейки, вызвав Мастер функций, в окне **Определенные пользователем** выбрав Gipotenuza и запустив на выполнение, получим необходимый результат.

Элемент управления `UserForm` (пользовательская форма) часто используют, чтобы собрать разные данные в одном месте, например создать проект для ответа на вопрос на разных листах Excel. Интерфейс и программный код для реализации проекта представлены на рис. 3.20.

Word в сочетании с VBA дает широкие возможности для работы с текстом. При работе в Word часто возникает задача доступа к определенной части документа. Для этого используют коллекции объектов: `Sections` (разделы), `Paragraphs` (абзацы), `Sentences` (предложения), `Words` (слова) и др. Эти коллекции имеют метод `Count`, позволяющий определить число объектов в коллекции. Например, `...Words (5)` — перейти к пятому слову в документе.

Часто используют следующие объекты.

`Font` — позволяет изменять свойства шрифта и другие объекты форматирования, поддерживает методы диалогового окна **Шрифт**, вызываемого из меню **Формат**.

Рис. 3.20. Интерфейс и программный код для проекта ответа на вопрос

Например, `Gal. Sentences(3).Font.Size= 20` — здесь в документе под именем «Gal» в третьем предложении размер шрифта 20 пунктов.

`Range` — базовый объект для редактирования, включающий слова и предложения, но не раздел и абзац. Например, `Gal. Paragraph(3).Range.InserAfte"Я учусь"` — в документе под именем «Gal» после третьего абзаца вставляется текст «Я учусь».

`Gal. Sentences(3).InserAfte"Я учусь"` — текст вставляется после третьего предложения.

`Range` поддерживает основные методы форматирования текста.

Часто возникает задача определения общего числа слов в документе. Тогда можно, создав кнопку и определив `n` как общее число слов, записать следующую процедуру.

```
Private Sub CmdStart_Click()
n=Thisdocument.Words.Count
For i=1 To n
Nomfont
Next
i=1
Vidfont
End Sub
```

Справочную систему VBA, находясь в среде программирования, можно вызвать несколькими способами:

- нажать на клавишу **F2**;
- выбрать команды меню **View** → **Object** → **Browser**;
- нажать кнопку **Help**.

В этой системе можно получить сведения об объектах, свойствах, методах и синтаксисе языка. При нажатии клавиши **F1** можно получить подробные сведения по выбранному слову.

Пример выполнения работы «Программирование в VBA на рабочем листе Excel и в документе MS Word»

1. Цель работы.

Цель работы — получить практические навыки по разработке программного кода и отладке на языке Visual Basic.

2. Исходные данные.

Вариант *N*.

N.1. Выполнить указанные задания с элементом управления `Label` (Надпись) (см. с. 202).

N.2. Сложить три числа.

N.3. Создать свой тип шрифта и использовать его для второго раздела активного документа (см. с. 205).

3. Перечень применяемых приборов и материалов.

3.1. ПЭВМ: Intel Celeron (33 МГц, 32 Мб).

3.2. Программное обеспечение: ОС Windows, система программирования VBA.

4. Ход и результаты работы.

4.1. Открываем MS Excel.

4.2. Запускаем среду VBA и открываем панель инструментов и панель свойств.

4.3. На рабочем листе Excel Лист1 размещаем два элемента `A` (`Label` — «Надпись»). Первый используем для вывода сообщения, второй — для размещения рисунков (рис. 3.21).

4.4. Устанавливаем курсор в правом столбце окна свойств и работаем с его основными свойствами `Label`: `Name`, `BackColor`, `BorderColor`, `Caption`, `Enabled`, `Font`, `ForeColor`, `Picture`, `Visible`, `Height`, `Width`, `Left`, `Top`.

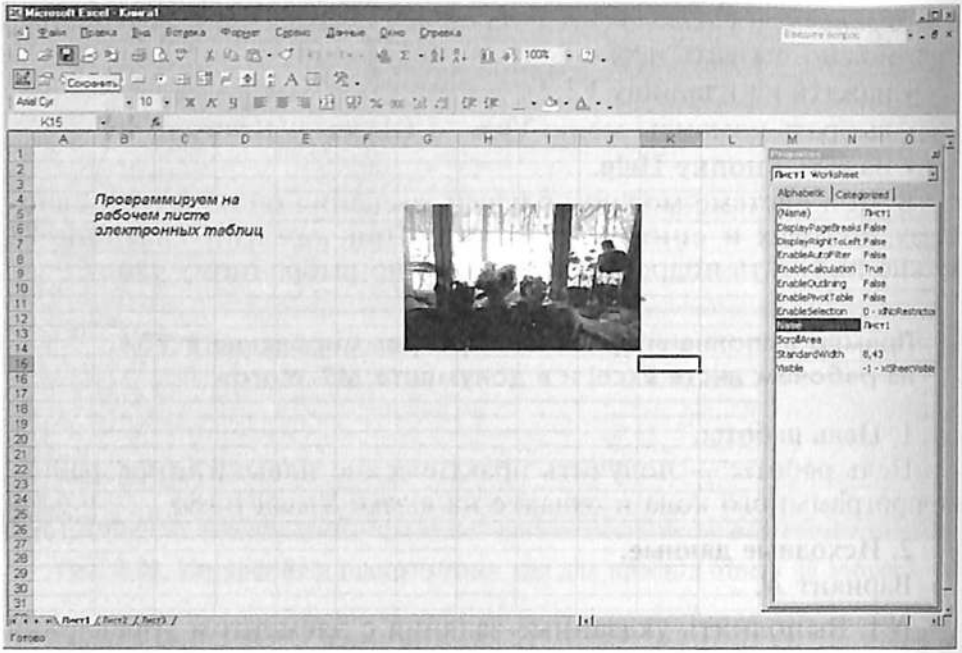


Рис. 3.21. Результат работы с элементом управления `Label`

4.5. Для решения задачи сложения трех чисел на листе 2 создаем командную кнопку, даем ей надпись «Действие», в свойстве `Name` задаем имя `Cmd1`. Два раза щелкнув по кнопке, выходим в редактор кода и создаем процедуру. В местной (`Private`) процедуре `SubCmd_Clik()`, где мигает курсор, набираем следующий код (рис. 3.22).

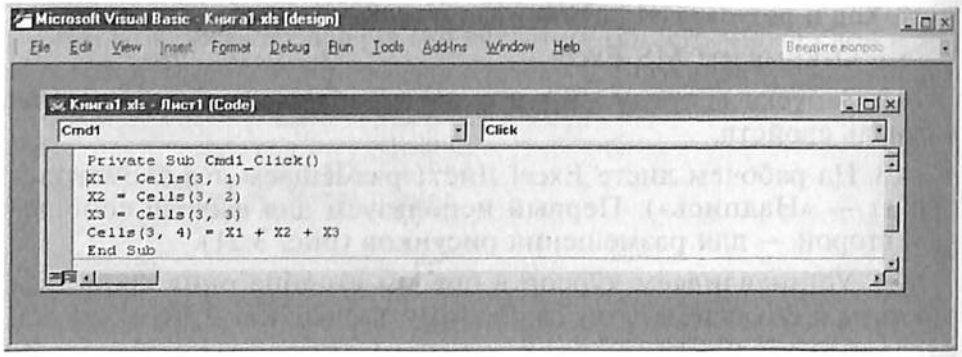


Рис. 3.22. Программный код для решения задачи сложения трех чисел

Между началом и концом процедуры код можно записать и так:

```
Cells (3,4)= Cells(3,1)+ Cells(3,2) + Cells(3,3)
```

Нажав кнопку в *левом* верхнем углу редактора, возвращаемся в электронную таблицу.

Теперь в ячейки A3, B3, C3 можно вводить любые числа и, нажав на кнопку **Действие**, в ячейке D3 получать соответствующий результат (рис. 3.23).

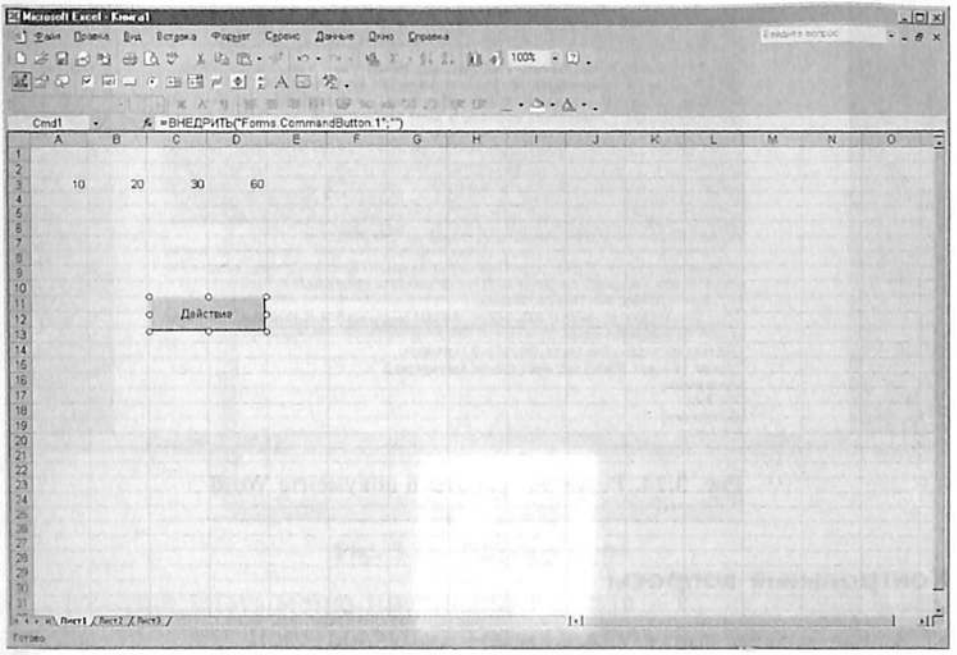


Рис. 3.23. Выполняемый проект для решения задачи

Сохраняем проект.

Для создания своего типа шрифта и использования его для второго раздела активного документа выполняем следующее.

Войдя из созданного документа Word в редактор кода VBA, пишем следующий код:

```
Set myFont = NewFont
myFont.Bold = True
myFontName = "Corbel"
ActiveDocument.Paragraphs(2).Range.Font=myFont
```

В результате запуска программы на выполнение получен следующий результат (рис. 3.24).

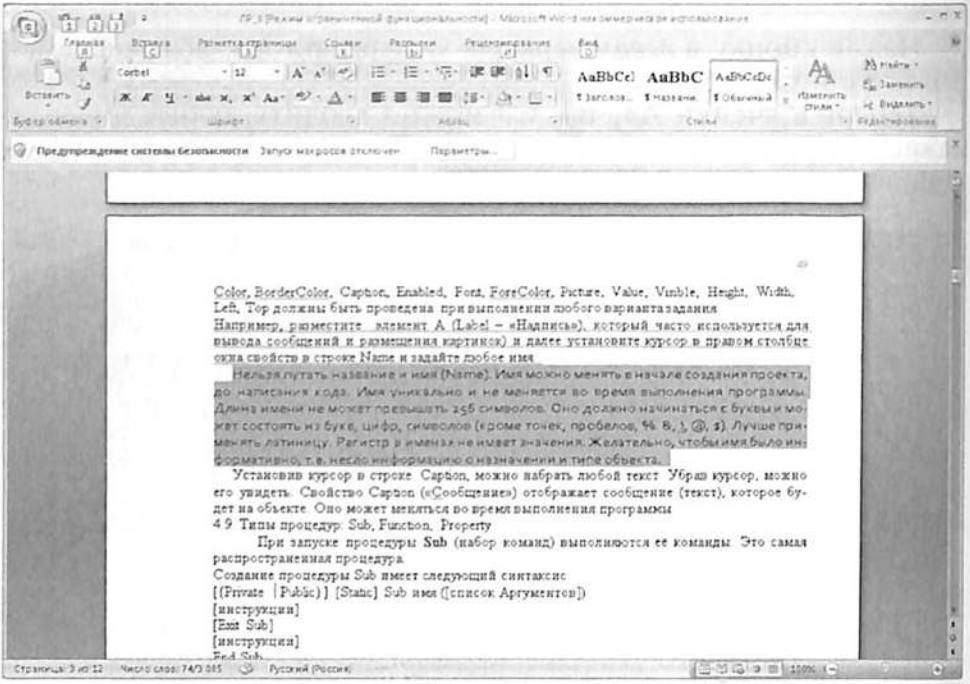


Рис. 3.24. Результат работы в документе Word

Контрольные вопросы

1. Как вызвать среду VBA в Excel?
2. Когда в Excel полезно применять VBA?
3. Какие кнопки содержит панель управления VBA?
4. Каково назначение объекта `Range`?
5. Каково различие между названием и именем (`Name`)?
6. Каковы типы процедур VBA?
7. Каковы возможности для работы с текстом в Word в сочетании с VBA?
8. Какими способами можно вызвать справочную систему VBA?

3.7.4. Создание меню в MS Visual Basic

Редактор меню (Menu Editor) позволяет добавлять в форму новые меню, изменять и удалять существующие, обогащать их специальными эффектами. Используя процедуры событий, можно задействовать команды меню.

Чтобы создать на форме меню, следует выбрать команды меню **Tools** → **Menu Editor** или на панели инструментов щелкнуть по кнопке **Menu Editor**. На экран будет выведено окно редактора меню (рис. 3.25).

Рис. 3.25. Редактор меню

Нижняя часть меню предназначена для всех элементов меню данной формы. Верхняя часть меню предназначена для свойств отдельного элемента меню. В поле **Заголовок** указываются название меню и название опций (команд меню). Если перед названием поставить знак &, то будет назначена комбинация клавиш доступа к меню или его опции. Например, `&File` означает, что меню может быть вызвано нажатием клавиш **Alt+F**.

Имя меню в программе состоит из префикса `mnu` и последовательности опций (например, `mnuFileSave` для опции **File/Save**) и задается в поле **Имя**.

После ввода названия и имени нужно щелкнуть мышью по кнопке **Следующий**. Имя меню появится в нижней части окна. Иерархию структуры меню определяют уровни отступов в спи-

ске элементов, которые являются именами меню. Опции меню указываются ниже имени меню и показываются с отступом. Если опция содержит подменю, то опции подменю будут указаны ниже с дальнейшим отступом (см. рис. 3.25).

Для создания опции меню нужно заполнить поля **Заголовок** и **Имя** и щелкнуть по стрелке, направленной вправо. Появившиеся четыре точки показывают, что это опция указанного выше меню. Опция меню может быть введена. Если введены все опции меню, то щелчком мыши по стрелке, направленной влево, следует удалить точки со следующей строки, позволяя ввести новое имя меню.

Создание подменю осуществляется путем добавления опций на втором уровне отступов. При этом опции первого уровня становятся именами подменю. Можно создать четыре уровня подменю.

Редактировать элемент меню можно, щелкнув по нему, выделив его и далее выполнив следующие действия:

- используя кнопку **Вставить**, можно вставить новую опцию выше выделенного элемента;
- используя кнопку **Удалить**, можно удалить выделенный элемент;
- щелчком мыши по стрелке, направленной влево, можно выделенный элемент из опции сделать именем меню, а по стрелке, направленной вправо, — наоборот;
- щелчком мыши по стрелке, направленной вверх, можно переместить выделенный элемент вверх по списку, а по стрелке, направленной вниз, — вниз по списку.

В редакторе меню есть много свойств для элемента меню. Три свойства представлены контрольными индикаторами и являются булевыми, т. е. могут принимать значение `True` или `False`. Это `Checked` (Контролируемость), `Enabled` (Доступность), `Visible` (Видимость).

Если для `Checked` установлено значение `True`, то появится галочка слева от опции меню. Щелчок мышью по опции меню устанавливает для `Enabled` значение `True` и помещает напротив него галочку. Удаляют галочку, повторно щелкнув по этой опции — для `Checked` установится значение `False`.

`Enabled` указывает доступность опции меню. Для этого свойства также используются значения `True` или `False`.

`Visible` используют для ограничения опций, чтобы, например, некоторые опции меню увидеть только с помощью паролей.

Клавишу быстрого доступа для любого элемента меню назначают с помощью щелчка по окну **Shortcut** и выбора варианта из раскрывающегося списка. В меню с большим количеством опций последние целесообразно разделить на группы по общим признакам с помощью линий раздела. Эти линии в меню находятся как элементы со свойством `Caption`, имеющим значение _ (подчеркивание), и имеют свои значения `Name`.

В приложениях могут быть полезны следующие свойства:

- `Index` — хранит индекс элемента в меню;
- `Help Context ID` — обеспечивает связь со справочным файлом;
- `Negotiate Position` — определяет положение меню, если на экран их выводится несколько;
- `Window List` — определяет, содержит ли меню список других открытых окон.

Для каждой опции (команды) меню необходимо написать процедуры событий, которые содержат программные утверждения вывода или обработки информации на форме и изменяют свойства меню. Общение с пользователем происходит через диалоговое окно с помощью одного из объектов ввода или объекта общих диалогов. Например, событие `Click` происходит при щелчке мыши по опции меню или при нажатии на клавиши доступа.

Буфер обмена, как и любой другой объект, имеет доступные ему методы, т. е. внутренние процедуры, которые выполняются из вызванной событием процедуры. Таким событием является щелчок мыши по команде меню. Для выполнения метода его нужно записать в следующем формате: `Object.method`.

Запись в процедуре строки `Clipboard.Clear` приводит к очистке буфера обмена. Буфер обмена содержит следующие методы:

- `Clear` — выполняет очистку содержимого буфера;
- `SetText` — копирует указанный текст в буфер обмена;
- `GetText()` — возвращает содержимое буфера обмена.

Методы буфера обмена могут быть использованы в сочетании со свойствами текстовых и графических окон.

Для получения практических навыков разработки меню в Visual Basic рекомендуется на компьютере выполнить следующий пример: создать меню **Edit** (Редактирование) с опциями копирования (**Copy**), вставки (**Paste**), удаления (**Clear**), удаления и копирования в буфер обмена (**Cut**).

Последовательность работы должна быть такой:

1. Запуск Visual Basic и в меню **Файл** — щелчок мышью по команде **Новый проект**.

2. Щелчок по кнопке **Редактор меню** (**Menu Editor**) на панели инструментов. На экране появится редактор меню.

3. Набираем Edit в поле **Заголовок** (название появляется в нижней части окна), и при нажатии клавиши **Tab** курсор перемещается в поле **Имя**. Для создания клавиши доступа к меню **Edit** в поле **Заголовок** перед буквой E вводится знак &.

Аналогичные действия выполняем для команд **Copy**, **Paste**, **Clear** и **Cut**.

4. Набираем текст `mnuEdit` в поле **Имя** (префикс `mnu` применяется для идентификации пункта меню в программном коде).

5. Щелчок по кнопке **Следующий** для добавления заголовка меню **Edit** в программу. Меню **Edit** добавится к строке меню, а диалоговое окно очистится. По мере создания меню каждый пункт будет добавляться в окно списка меню. Структура создаваемого меню видна.

6. Набираем текст Copy в поле **Заголовок**, а после нажатия клавиши **Tab** в поле **Имя** — `mnuCopyItem`. Используется соглашение о наименовании: в конец имени добавляется слово `Item`. Команда **Copy** появится в списке меню.

7. Далее в списке меню щелчок по кнопке со стрелкой, направленной вправо. Команда **Copy** переместится вправо, и перед ней появятся четыре точки, т. е. данный пункт стал командой меню.

8. Далее щелкаем по кнопке **Следующий** и для каждой из команд **Paste**, **Clear** и **Cut** осуществляем те же действия, что и для **Copy**.

9. Закрыв окно редактора меню щелчком по кнопке **OK**, увидим, что в среде программирования появилась форма со строкой меню **Edit** и опциями **Copy**, **Paste**, **Clear** и **Cut**.

Процесс создания меню **Edit** иллюстрирует рис. 3.26. Результат представлен на рис. 3.27. Список опций появляется при щелчке по **Edit**.

Методы буфера обмена были использованы в сочетании со свойствами текстовых окон. Например, для редактирования текста можно использовать следующие свойства текстового окна:

- `SelText` — содержит строку подсвеченных символов (пусто, если строка не подсвечена);

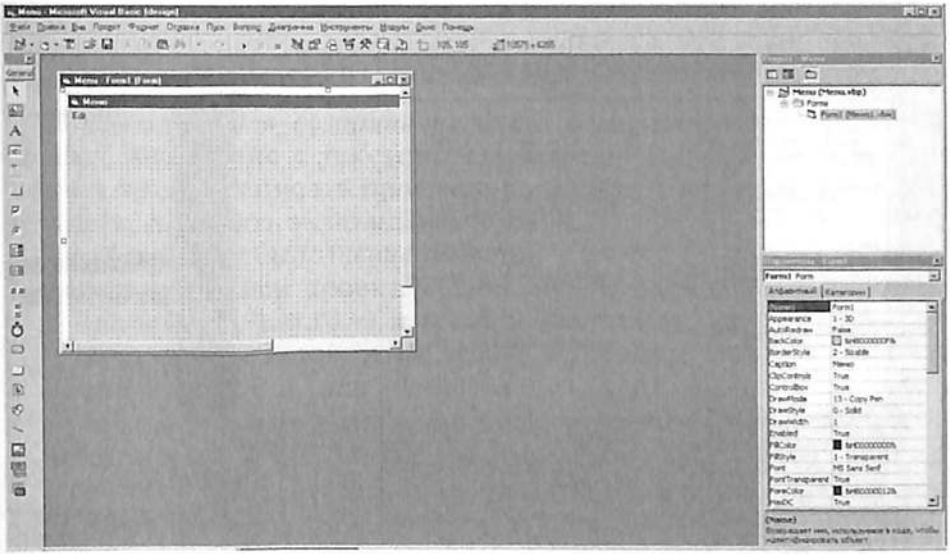


Рис. 3.26. Создание меню **Edit**

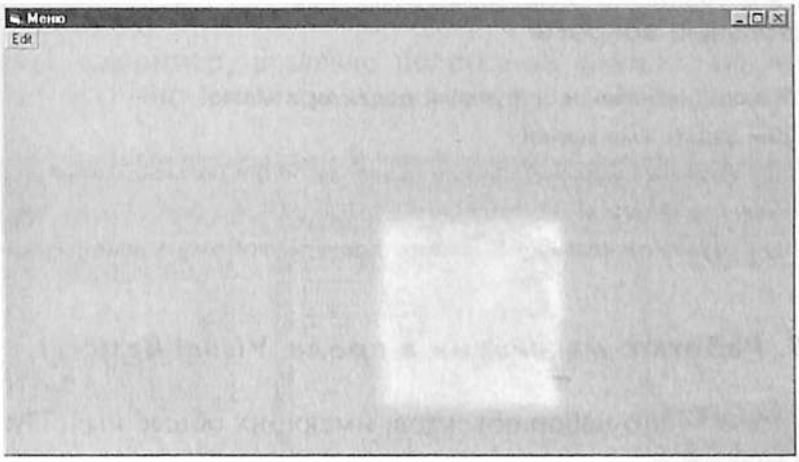


Рис. 3.27. Окончательный вид меню **Edit**

- `SelLength` — содержит длину (число символов) подсвеченной строки;
- `SelStart` — содержит текущее положение курсора.

Каждая команда меню выполняется отдельной процедурой по событию `Click()` — щелчок мышью по соответствующей команде меню. Следует написать следующие процедуры (рис. 3.28).

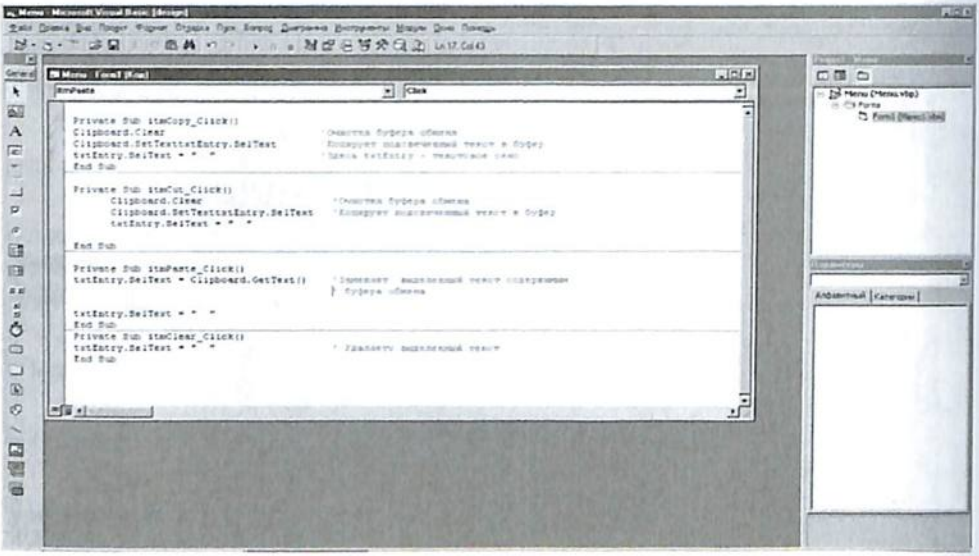


Рис. 3.28. Программный код

Контрольные вопросы

1. Каковы назначение и функции редактора меню?
2. Как задать имя меню?
3. Как создать название меню и названия опций (команд меню)?
4. Какие свойства меню вы знаете?
5. Как назначить клавишу быстрого доступа любому элементу меню?

3.7.5. Работа с массивами в среде Visual Basic

Массив — это набор объектов, имеющих общее имя. См. также раздел 3.5.

В зависимости от типа хранимых данных массивы могут быть любого типа — и текстовые, и числовые. Но все элементы массива имеют один и тот же тип. Массивы могут быть одномерные и многомерные.

Таблицы с одной строкой данных называются линейными. В программировании их называют *одномерными массивами*, так как положение элемента в таблице определяется одной мерой — его порядковым номером.

Пример 3.5.

1. *Постановка задачи*: заполнить массив случайными числами.

2. *Сценарий для создания проекта и его интерфейса*: создать проект, интерфейс с текстовым полем для вывода массива случайных чисел, написать программу, запустить ее на выполнение, отладить, получить исполняемый файл.

3. *Поэтапное выполнение работы*:

- создать новый проект. Разместить на форме текстовое поле (например, `txtDim`) и кнопку (например, `cmdDim`), дать ей заголовок «Заполнить», а для текстового поля установить значение `True` для свойства `MultiLine` (рис. 3.29, левое окно);
- написать в редакторе кода для кнопки `cmdDim` (Заполнить) событийную процедуру заполнения массива и его вывода в поле `txtDim` (рис. 3.29, центральное окно);
- запустить проект на выполнение и отладить его;
- в готовом проекте несколько раз щелкнуть по кнопке **Заполнить**. В поле `txtDim` появится массив случайных чисел (рис. 3.30).

Если, например, в задаче поставлено условие заполнения массива числами, вводимыми с клавиатуры, и вывода их на эк-

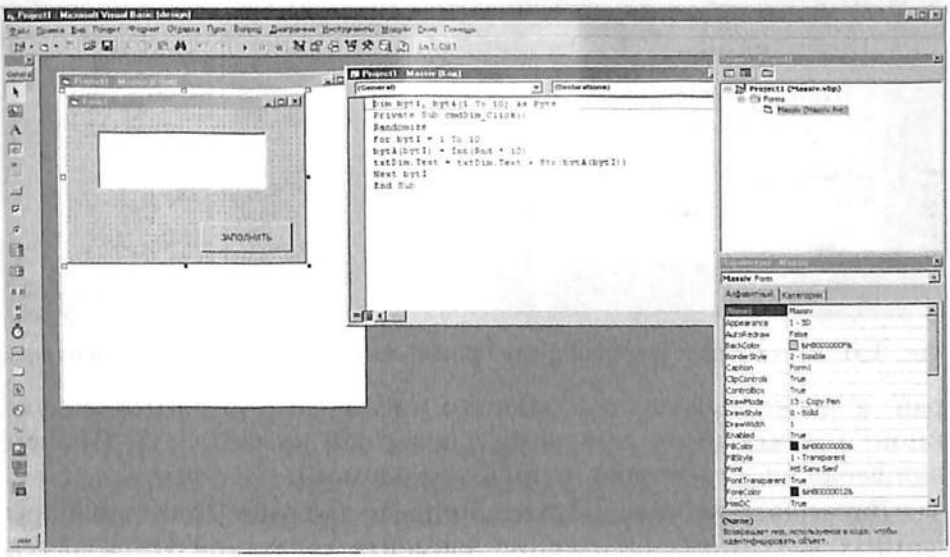


Рис. 3.29. Интерфейс и программный код проекта

Рис. 3.30. Массив случайных чисел

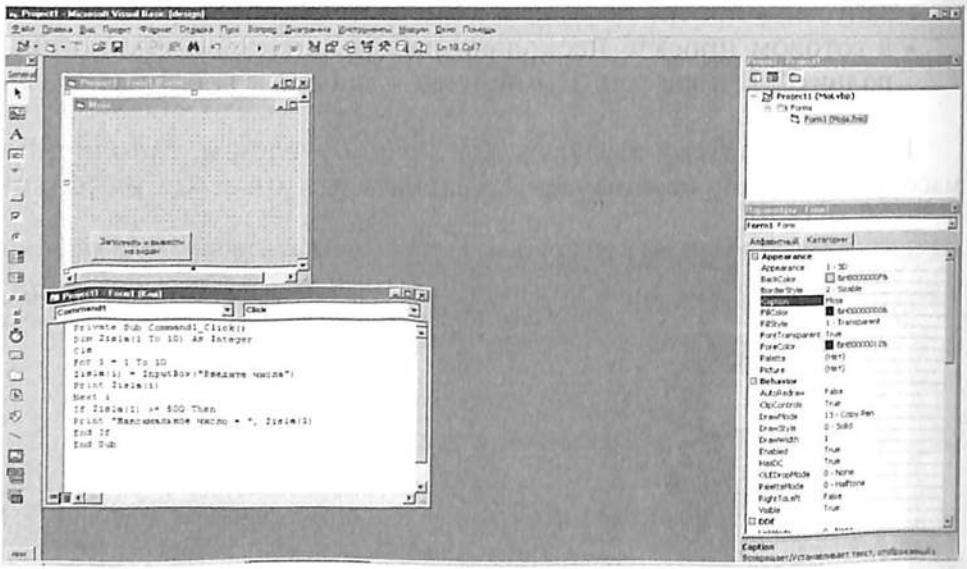


Рис. 3.31. Интерфейс и программа для проекта заполнения массива с клавиатуры

ран, а также вывода наибольшего числа, то программа модуля кнопки может иметь вид, представленный на рис. 3.31. Причем для ввода чисел массива используем элемент `InputBox`.

Двумерные массивы. Прямоугольные таблицы хранят наборы данных, в которых положение элемента характеризуется двумя индексами (положение по горизонтали и по вертикали).

Пример 3.6.

1. *Постановка задачи*: заполнить двумерный массив (4,5) целыми случайными числами, лежащими в интервале от 1 до 100, и вывести массив на экран в виде таблицы.

2. *Сценарий для создания интерфейса*:

а) создать таблицу из 5 колонок и 4 строк. Тип данных — целые. Вид цикла — со счетчиком;

б) записать алгоритм сохранения таблицы в массиве (4,5). Первый индекс (5) указывает число колонок, второй (4) — число строк.

3. *Алгоритм*:

```
АЛГ  (арг  цел  таб (4,5), рез цел таб)
НАЧ  случайные цел i,j
        нц
        для i от 1 до 4
        для j от 1 до 5
        X(I, J) = Int(Rnd(1) * 100) + 1
        кц
        вывод (таб)
КОН
```

4. *Поэтапное выполнение работы*:

- создать новый проект. Разместить на форме кнопку, дать ей название «Заполнить двумерный массив» (рис. 3.32);
- написать в редакторе кода для кнопки следующую событийную процедуру, причем внутренний цикл следует завершить точкой с запятой, что позволяет отображать массив построчно. Оператор `Print`, стоящий после внутреннего цикла, позволит переводить курсор на новую строку после вывода каждой строки.

```
CLS:  Randomize
Dim X(1 To 4, 1 To 5)
For I = 1 To 4
For J = 1 To 5
X(I, J) = Int(Rnd(1) * 100) + 1
Print X(I, J);
Next J
Print
Next I
```

- запустить проект на выполнение и отладить его;
- в готовом проекте несколько раз щелкнуть по кнопке **Создание**. В поле проекта появится массив двумерной таблицы (рис. 3.33).

Трехмерный массив можно представить в виде куба и т. д.

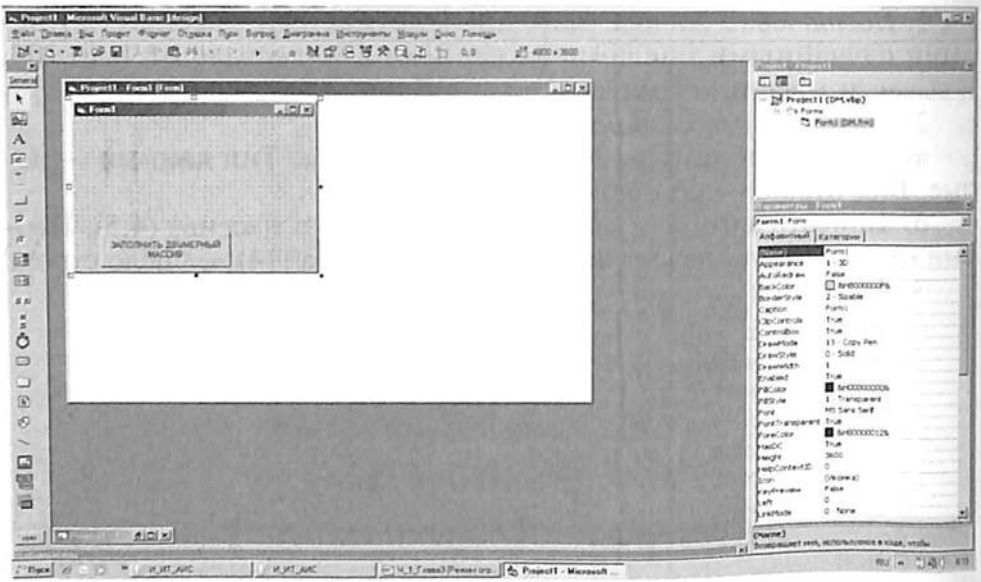


Рис. 3.32. Интерфейс для вывода двумерного массива

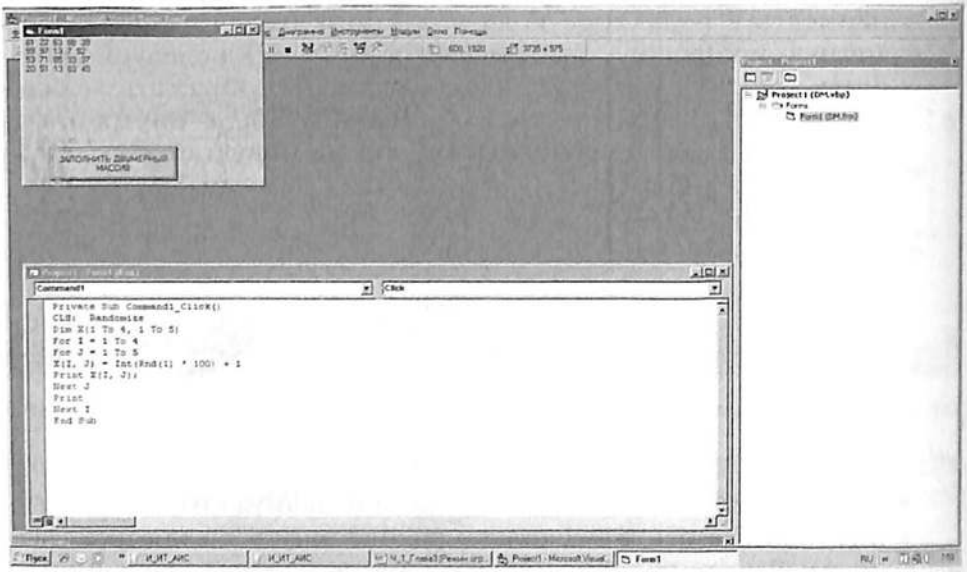


Рис. 3.33. Программный код и проект для создания двумерного массива случайных чисел

Контрольные вопросы

1. Что такое массив?
2. Что такое индекс массива?
3. Каковы способы присвоения значений элементам массива?
4. Какие виды массивов вы знаете, их характеристики?
5. Каково назначение оператора Dim?

3.7.6. Создание проектов с графикой, анимацией и звуком в MS Visual Basic 6.0

Для управления графикой используют следующие объекты управления:

- Image (Управление изображением) — предоставляет возможность добавлять в форму окно рисунка. Основные свойства объекта: Stretch (Вытягивание) — для определения точного размера рисунка; Picture (Рисунок) — для указания имени графического файла; Visible (Видимость) — для определения состояния рисунка;
- PictureBox (Графическое окно) — выводит на экран растровый файл (.bmp), файл пиктограммы (.ico) или метафайл (.wmf);
- Line (Отрезок) — предоставляет возможность создания линии на форме. Начало линии задается свойствами X1, Y1, а конец — X2, Y2 в типах относительно левого верхнего угла формы. Важные свойства объекта: BorderWidth (Толщина), BorderStyle (Тип линии), BorderColor (Цвет) и Visible (Видимость);
- Shape (Фигура) — предоставляет возможность создания различных фигур. Вид фигуры устанавливается значением свойства Shape: 0 — прямоугольник; 1 — квадрат; 2 — овал; 3 — круг, 4 — прямоугольник со скругленными углами; 5 — квадрат со скругленными углами. Для определения положения и размера фигуры используют свойства Left, Top, Width и Height. Свойства BorderColor, BorderStyle и BorderWidth определяют внешний вид линии, описывающей фигуру. Внутренний цвет и узор определяют свойства FillColor и FillStyle. Растровое изображение из внешнего файла можно добавить непосредственно на форму или объекты (элементы) управления PictureBox и Image.

Добавление растрового изображения на форму выполняется в следующем порядке:

1. Создать новый проект и присвоить имена ему и форме. Дать заголовок форме (в свойстве `Caption`).

2. В форму добавить объекты `PictureBox` и `Image`, присвоить им имена.

3. Установить положение и размеры для `PictureBox` и `Image` (рис. 3.34).

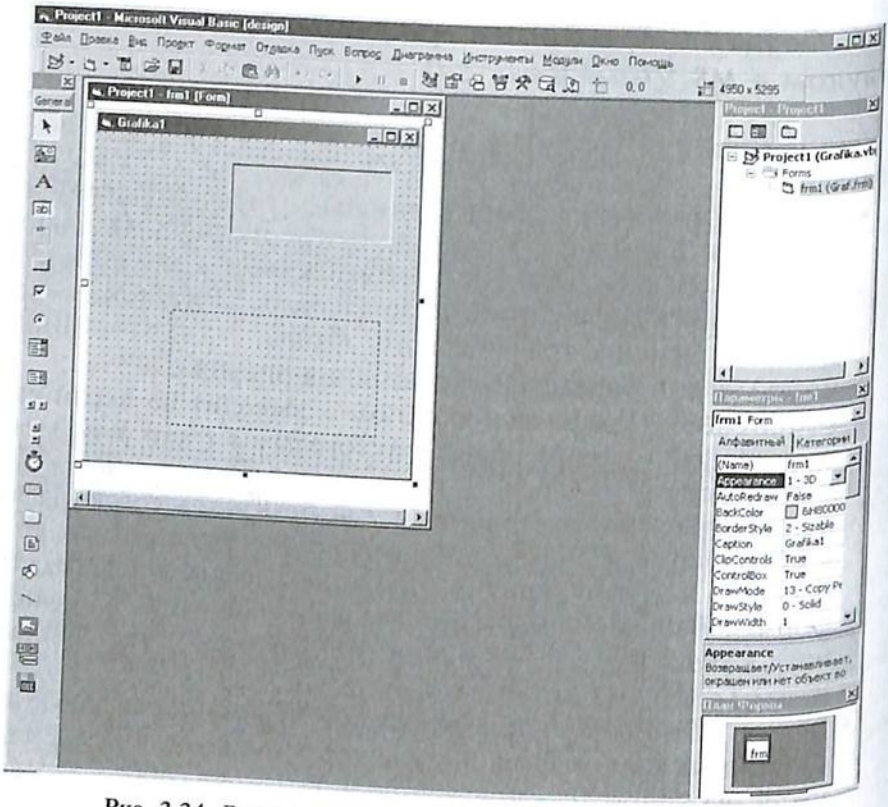


Рис. 3.34. Добавление растрового изображения на форму

4. В окне свойств **Параметры** формы щелкнуть мышью по свойству `Picture`. В открывшемся диалоговом окне **Load Picture** (Загрузка набора) в каталоге файлов выбрать файл с расширением .bmp и открыть его. На форме появится изображение (рис. 3.35).

5. Аналогично выполняются действия для элементов управления `PictureBox` и `Image`. Размеры изображения соотносятся с размерами окна с помощью свойства `AutoSize` объекта.

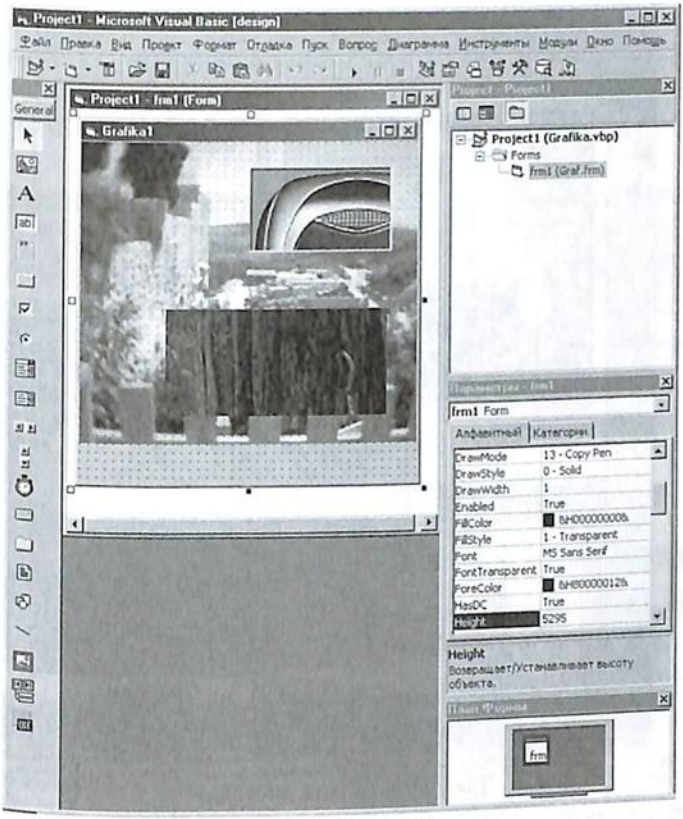


Рис. 3.35. Изображение на форме

6. Сохранить проект и запустить программу.

Для смены рисунков нужно изменять значение свойства `Picture` в окне **Image**. Для этого в форму нужно добавить кнопку с названием (например, `cmdChange`), а в процедуру обработки события `cmdChange_Click ()` добавить код (рис. 3.36). Результат работы программы показан на рис. 3.37.

Для добавления рисунков на форму во время выполнения программы нужно использовать функцию `LoadPicture()`, имеющую следующий синтаксис:

```
My Picture = LoadPicture(strFilePath)
```

где `MyPicture` — имя элемента управления, данное пользователем;

`strFilePath` — строковая переменная, содержащая полный путь к графическому файлу.

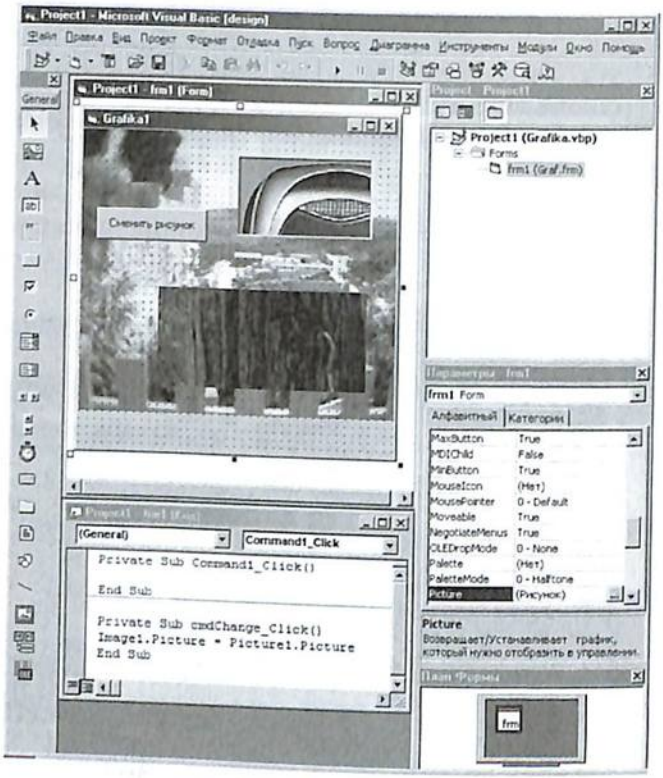


Рис. 3.36. Смена рисунка

Рис. 3.37. Результат работы программы

Элементы **Line** и **Shape**, находящиеся на панели инструментов, позволяют создавать простые рисунки (рис. 3.38). Но эти рисунки не могут использоваться программой во время выполнения и отображаться поверх других объектов.

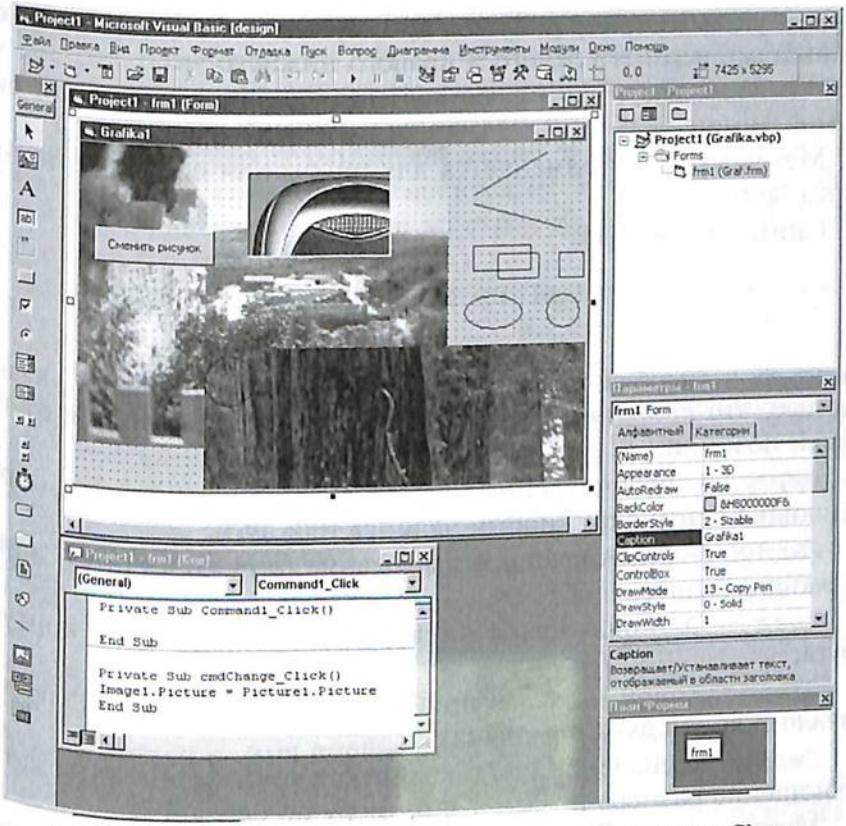


Рис. 3.38. Рисунок, созданный с помощью элементов **Line** и **Shape**

Создать изображения можно с помощью графических методов. Чаще других используются методы `Line`, `Circle`, `PSet`.

Метод `Line` применяется для рисования отрезков, прямоугольников и многоугольников. Синтаксис метода:

```
Object.Line [Step] (x1, y1) [Step] ] (x2, y2), [цвет], [B] [F]
```

где `Object` — имя объекта, в котором создается рисунок; если оно отсутствует, рисунок создается на форме;

`Line` — название метода;

Step — ключевое слово, указывающее начальную позицию отсчета координат;

(x1, y1) и (x2, y2) — координаты соответственно начальной и конечной точек изображаемой линии;

цвет — длинное целое число, определяющее цвет линии;

B — указывает, что рисуется прямоугольник с заданными координатами верхнего левого и правого нижнего угла;

F — указывает, что прямоугольник заполняется цветом рисуемой линии.

Метод Circle применяется для рисования окружности, эллипса, дуги.

Синтаксис метода:

```
Object.Circle, [Step] (x, y), Радиус, [Цвет, Начало,
Конец, Сжатие]
```

где Step — (необязательный параметр) ключевое слово, указывающее, что координата центра окружности вычисляется от текущей позиции;

(x, y) — целочисленные выражения или константы, определяющие центр окружности, эллипса или дуги;

Радиус — целочисленное выражение (или константа), определяющее радиус;

Цвет — длинное целое число, указывающее цвет контура круга;

Начало, Конец — выражения или константы, определяющие начало и конец дуги в радианах;

Сжатие — значение, устанавливающее коэффициент сжатия (отношение размера по оси *Y* к размеру по оси *X*).

Метод PSet применяется для установки цвета отдельного пикселя на экране.

Синтаксис метода:

```
Object.PSet [Step] (x, y), [Цвет]
```

где (x, y) — выражения или константы, определяющие координаты для установки цвета;

Цвет — длинное целое число, определяющее цвет отметки. Если опущено, то используется текущая (актуальная) установка свойства ForeColor. Чтобы определить цвет, можно использовать функцию RGB или QBColor.

Свойство BackColorv можно использовать, чтобы очистить одиночный пиксель, указав его координаты.

Анимация — иллюзия движения графического объекта на экране. Для создания анимации используется принцип смены кадров изображения. Имитация движения объекта осуществляется программно. В программе предусматривается написание кода, создающего изображение, фиксирующего его в определенный момент времени, изменяющего изображение. Чтобы манипулировать скоростью объекта, используют пустой цикл, замедляющий движение. Программно движение задается координатами или смещением координат относительно начала системы координат экранной формы (сетки из строк и столбцов), находящегося в верхнем левом углу формы.

Синтаксис оператора перемещения имеет следующий вид:

```
ObjectName.Move x, y
```

где ObjectName — имя перемещающегося по форме объекта;

Move — команда перемещения;

x, y — координаты нового положения объекта на форме.

Командами Left (Слева) и Top (Сверху) можно задать смещение координат относительно текущего положения объекта.

Синтаксис оператора смещения имеет следующий вид:

```
ObjectName.Move ObjectName. Left +/- x
ObjectName. Top +/- y
```

где ObjectName — имя смещаемого на форме объекта;

+/-x — знаки, указывающие направление смещения.

Перемещение объекта в форме показано далее в примере.

Создание *звуковых эффектов* можно осуществить с помощью оператора Beep (Звуковой сигнал). Например, для создания двух последовательных однотонных сигналов нужно записать:

```
Dim I
For I = 1 To 2
     Beep
Next I.
```

Для создания более сложных звуковых эффектов нужно использовать специальные программы, например MS Multimedia Control.

Пример выполнения работы «Создание проектов с графикой, анимацией и звуком в MS Visual Basic 6.0»

1. **Цель работы.**

Получить практические навыки разработки проекта с графикой, анимацией и звуком в MS Visual Basic.

2. **Исходные данные.**

Графический файл с расширением .bmp, содержащий рисунок.

3. **Используемые приборы и материалы:**

3.1. Персональный компьютер — ПЭВМ.

3.2. Программное обеспечение — операционная система Windows XP, ППП MS Office, система программирования на языке Visual Basic — MS Visual 6.0 и выше.

4. **Ход и результаты работы.**

4.1. Работа предусматривала создание проектов с графикой, анимацией и звуком в MS Visual Basic 6.0.

4.2. Разработаны проекты, представленные на рис. 3.39—3.43. Последовательность работы приводится ниже.

4.2.1. Запуск Visual Basic. В меню **File** щелчок мышью по команде **New Project**. Создан новый проект типа Standard.exe, и ему присвоено имя Graf.

4.2.2. Форме присвоено имя `my1` и дан заголовок `Graf1`.

4.2.3. В форму добавлен объект `Image`, и установлено для него имя `Image1`.

4.2.4. В окне свойств **Параметры** формы — щелчок мышью по свойству `Picture`. В открывшемся диалоговом окне **Load Picture** (Загрузка рисунка) в каталоге файлов выбран файл с расширением .bmp и открыт (рис. 3.39).

4.2.5. На форме появилось изображение (рис. 3.40). Аналогичные действия выполнены для формы. Размеры изображения соотнесены с размерами окна с помощью свойства `AutoSize`.

4.2.6. С помощью элементов управления **Line** и **Shape** созданы простые рисунки (рис. 3.41).

4.2.7. Проект сохранен, и программа запущена на выполнение.

4.3. Более сложные изображения (концентрические окружности) созданы с помощью графического метода `Circle`.

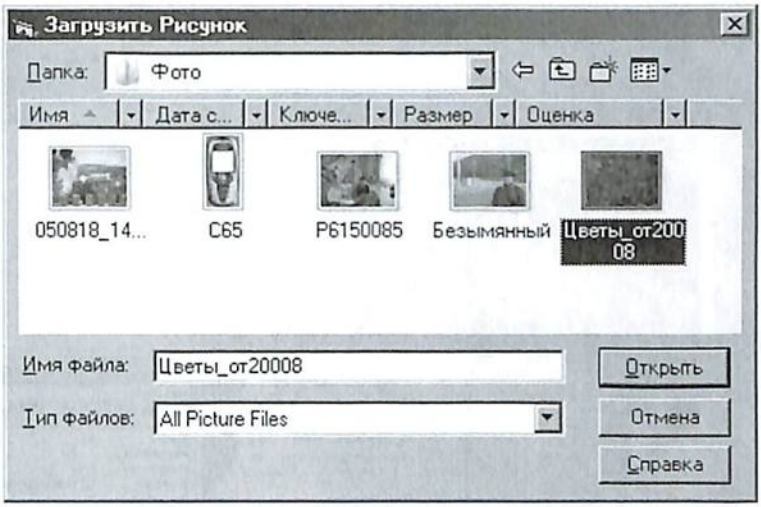


Рис. 3.39. Выбор файла с рисунком

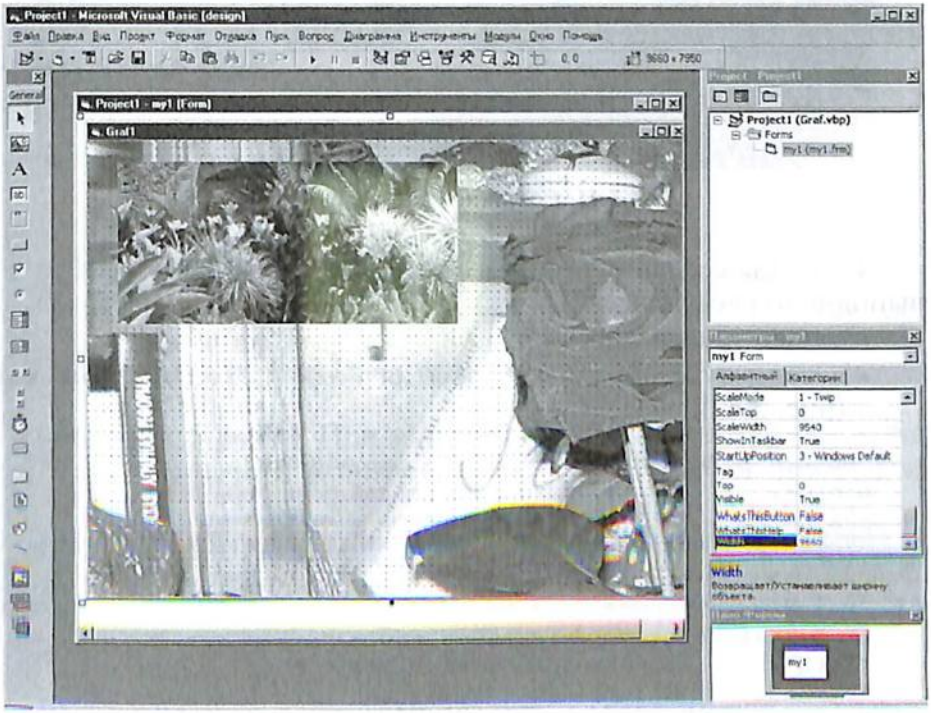


Рис. 3.40. Изображение на форме

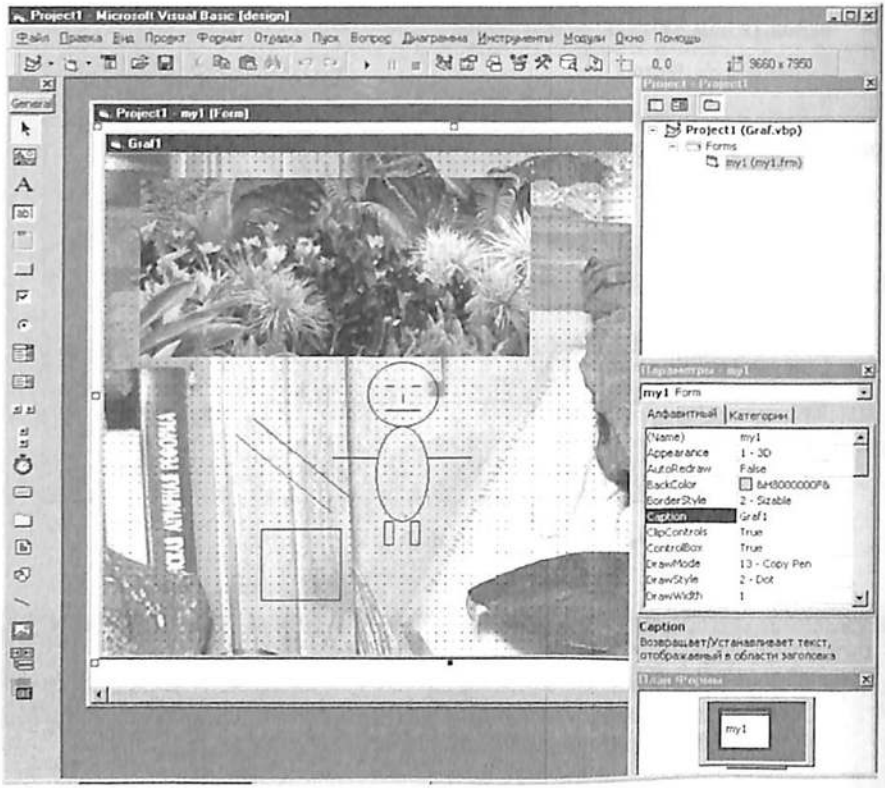


Рис. 3.41. Создание простых рисунков

4.3.1. Для создания программы, использующей метод `Circle`, выполнено следующее.

- создан новый проект;
- в процедуре обработки события `Form_Ctick` написан код (рис. 3.42).

```
'Разноцветные 'концентрические окружности
Dim CX, CY, Radius, Limit 'Объявление переменных
ScaleMode = 3 'Установка масштаба в пикселях
'Установка Координаты X центра окружностей в центре формы
CX = ScaleWidth / 2
'Установка координаты Y центра окружностей в центре формы
CY = ScaleHeight / 2
If CX > CY then Limit = CY Else Limit = CX
For Radius = 0 To Limit  'Установка значения радиуса
For I = 1 To 10000       'Вспомогательный цикл-задержка,
                         ' необязателен
```

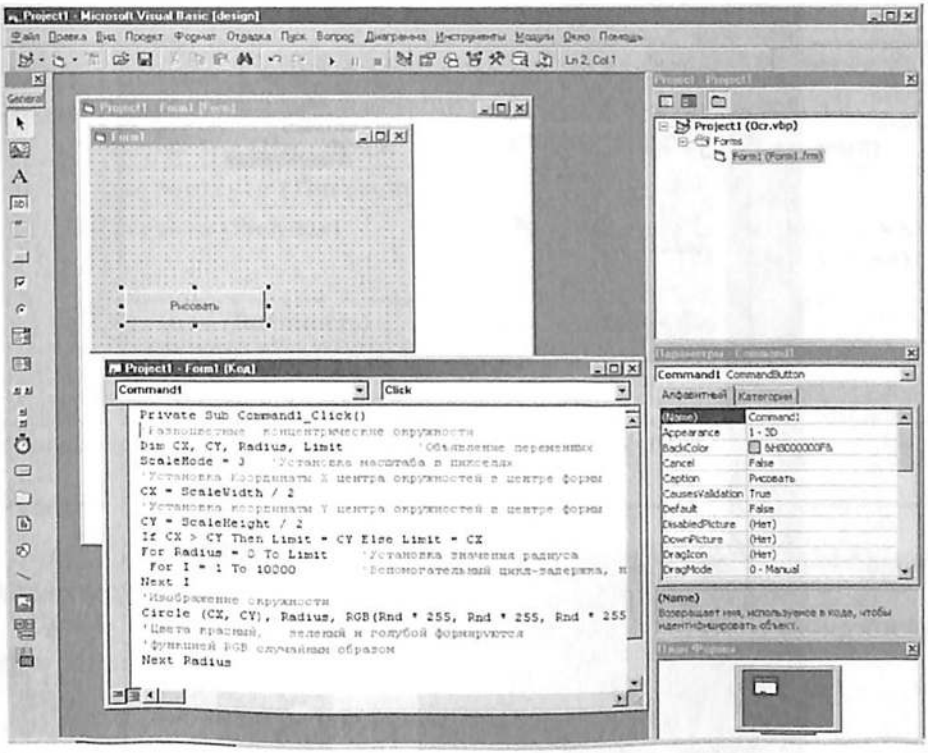


Рис. 3.42. Код процедуры для изображения концентрических окружностей

```
Next I
'Изображение окружности
Circle (CX, CY), Radius, RGB(Rnd * 255, Rnd * 255, Rnd *
255)
'Цвета красный, зеленый и голубой формируются
'функцией RGB случайным образом
Next Radius
```

Проект сохранен. При запуске на выполнение программа начинает изображать концентрические окружности различного цвета.

4.4. Проект с *эффектами анимации* (перемещение объекта по окружности) был создан с помощью команды `Move` и объекта типа `Таймер`, который устанавливал скорость движения объекта в форме. Последовательность работы приводится ниже.

4.4.1. Создан новый проект.

4.4.2. В форму помещены объекты управления `Image`, `Timer` и два объекта управления `CommandButton` (рис. 3.43).

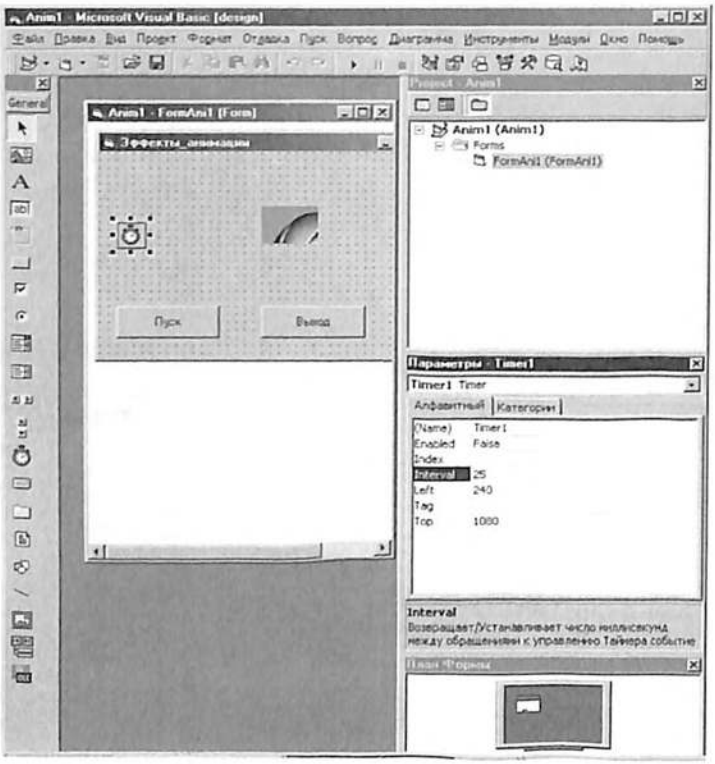


Рис. 3.43. Форма с размещенными в ней объектами

4.4.3. Для рисунка и таймера установлены свойства в соответствии с табл. 3.3.

4.4.4. Свойству `Caption` объекта `Command1` присвоено значение `Пуск`, а свойству `Caption` объекта `Command2` — `Выход`.

4.4.5. В разделе `"General"` объявлены переменные:

```
'PI - число Pi; XI, Yl - текущие координаты,
'Alf - значение угла,
'Rad - радиус окружности
Option Explicit
Dim PI As Single
Dim XI, Yl, Alf, Rad As Integer
Dim Da AS Double
```

4.4.6. В процедуре обработки события `Command1_Click` написан программный код:

```
Timer1.Enabled = True     'Активизация таймера
```

Таблица 3.3. **Установка значений свойств объектов `Image1` и `Timer1`**

Объект	Свойства	Устанавливаемое значение
`Image1`	`Appearance` (Внешний вид)	`0-Flat` (Плоский)
	`BackColor` (Цвет фона)	`light gray` (Светло-серый)
	`BorderStyle` (Тип рамки)	`0-None`(Нет)
	`Picture` (Рисунок)	`C:\ProgramFiles\DevStudeo\VB\Graphics\Icons\Arrows\Pomt01`
	`Visible` (Видимость)	`True` (Истина)
	`Left` (Относительная координата *x*)	`2160`
	`Top` (Относительная координата *y*)	`1800`
`Timer1`	`Enabled` (Включен)	`False` (Ложь)
	`Interval` (Интервал)	`25`

4.4.7. В процедуре обработки события `Command2_Click` написан код: `End`

4.4.8. В процедуре обработки события `FormLoad()` написан программный код:

```
'Инициализация начальных значений переменных
Rad = Form1.Height /50
Da = 0.1
Alf = 0
PI = 3.14
```

4.4.9. Открыт список **Object (Объект)** в окне **Code**, затем — щелчок мышью по объекту `Timer1`. В окне кода появилась процедура `Timer_Timer`. В нее записаны следующие операторы:

```
'Вычисление новых координат объекта
Y1 = Rad * Sin(Alf)
X1 = Rad * Cos(Alf)
'Проверка, не выходит ли изображение объекта за границы
If Image1.Left < (Form1.Width - Image1.Width) And_
(Image1.Top < _
Command1. Top - Image1 .Height) Then     ' Перемещение
                                  'объекта в новую позицию
Image1.Move Image1.Left - X1, Image1.Top -Y1
Alf = Alf + Da
Else
Timer1.Enabled = False       ' Выключение таймера
End If
```

Создание звуковых эффектов в виде трех однотонных последовательных сигналов было осуществлено с помощью оператора Beep. Запись кода имела следующий вид:

```
Dim I
For I = 1 To 3
     Beep
Next I.
```

Контрольные вопросы

1. Как добавить на форму растровые изображения с помощью элементов управления Image и PictureBox и выполнить смену рисунков?
2. Как создать простые изображения?
3. Каковы графические методы создания изображений?
4. Что такое анимация?
5. Как создать проект с анимационным объектом и звуком?

Часть II
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Глава 4
ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

4.1. Информация и информационные технологии. Сферы применения

Информация (от лат. *informatio* — разъяснение, изложение) — первоначально означала сведения, передаваемые людьми устным, письменным или другим способом с помощью условных сигналов, технических средств и т. д. Информация повышает уровень осведомленности человека об окружающем мире,

С середины XX в. информация — общенаучное понятие, включающее обмен сведениями между людьми, человеком и автоматом, автоматом и автоматом, обмен сигналами в животном и растительном мире; передачу признаков от клетки к клетке, от организма к организму, одно из основных понятий кибернетики.

Слово «информация» переводится на русский язык как сведения или сообщения. Эти сведения могут передаваться и фиксироваться в виде данных последовательностью различных символов, например букв русского и других алфавитов, цифр, знаков пунктуации, арифметических действий и др. Вместе с тем известно, что любые сообщения можно передавать и фиксировать путем их кодирования с использованием только двух символов, например точки и тире в азбуке Морзе, нуля и единицы в ЭВМ.

Физическая среда, в которой может фиксироваться или накапливаться информация для последующего прочтения, анализа и обработки, называется носителем информации.

Виды информации:

1) по способу восприятия: визуальная, аудиальная, тактильная, обонятельная, вкусовая;

2) по форме представления: текстовая, числовая, графическая;

3) по общественному значению: массовая, обыденная, общественно-политическая, эстетическая;

4) специальная, научная, производственная, личная.

Информация является первичным понятием. Можно утверждать, что это понятие предполагает наличие материального носителя информации, источника информации, передатчика информации, приемника и канала связи между источником и приемником. Особенность этого понятия в том, что оно используется во всех без исключения сферах: в философии, естественных и гуманитарных науках, биологии, медицине и физиологии, психологии человека и животных, социологии, искусстве, в технике и экономике, в повседневной жизни. Поэтому конкретное толкование элементов, связанных с понятием «информация», зависит от метода конкретной науки, цели исследования или просто от наших житейских представлений.

Рассуждая о количестве (много информации, мало информации), мы не можем сказать, каков объем (или количество) полученной информации. С точки зрения компьютера ответ прост: один бит (да или нет, 1 или 0). Но человек не компьютер, и для него объем полученной информации связан с «коэффициентом неожиданности», который, в свою очередь, зависит от предварительных знаний человека. Объем полученной информации изменяется в зависимости от вероятности события, которое тоже зависит от множества факторов.

Данные — см. раздел 1.2.

Различие понятий «информация» и «данные» в том, что с информацией имеет дело человек, которого интересует ее содержание, смысл, а с данными имеет дело, как правило, техническая система, которая их обрабатывает безотносительно к содержанию, смыслу. Информация кодируется с помощью данных.

Человек перерабатывает информацию минимум на трех уровнях: на физиологическом уровне (с помощью органов чувств), на уровне рационального мышления, на уровне подсознания. Про-

цесс переработки крайне сложен, он зависит от жизненного опыта человека, от эрудиции, от профессии, от заинтересованности в тех или иных сведениях и др.

Особая проблема — процесс выработки человеком новой информации (научной или художественной). Новая информация полезна не только для общего развития, она помогает понять, как соотносятся между собой процессы обработки информации человеком и компьютером и как сам человек связан с мировым информационным пространством. В процессе получения новой информации речь уже идет о получении знаний.

Знания — см. раздел 1.2. Знания бывают эмпирические, теоретические, житейские, донаучные, научные и др. (см. также главу 6).

Информационный процесс — процесс, в результате которого осуществляется прием, передача, преобразование, защита, поиск, хранение и использование информации. Информационные процессы: поиск, сбор, хранение, обработка, передача, использование, защита информации.

Технология — это комплекс научных и инженерных знаний, воплощающихся в приемах труда, наборах материальных, технических, энергетических, трудовых факторов производства, способов их соединения для создания продукта или услуги, отвечающих определенным требованиям или стандартам.

Информационная технология — см. раздел 1.2.

Цель информационной технологии — производство информации для ее анализа и принятия на его основе решения по выполнению какого-либо действия.

В современном обществе основным техническим средством обработки информации служит персональный компьютер. Внедрение ПК в информационную сферу и использование телекоммуникаций определили новый этап развития информационной технологии, которая с этого момента получает наименования «новая», «компьютерная». Определение «новая» подчеркивает ее радикально новаторский, а не эволюционный характер. Внедрение ее существенно изменяет содержание различных видов деятельности в учреждениях и организациях. В сферу новой информационной технологии включены также коммуникационные технологии, обеспечивающие передачу информации различными средствами, такими как телефон, телеграф, телевидение, факс и др. Определение «компьютерная» подчеркивает, что основным техническим средством ее реализации является компьютер. Су-

ществует три основных принципа компьютерной информационной технологии:

- интерактивный (диалоговый) режим работы с компьютером;
- интеграция с другими программными продуктами;
- гибкое изменение данных и поставленных задач.

Технологический процесс материального производства реализуют с помощью различных технических средств: оборудования, станков, инструментов, конвейерных линий и т. д. По аналогии в информационной технологии в роли технических средств производства информации выступает аппаратное, программное и математическое обеспечение этого процесса. С их участием первичная информация перерабатывается в информацию нового качества.

Инструментарием информационной технологии является совокупность программных продуктов, которые применяются для достижения поставленной пользователем цели. Все известные программные продукты общего назначения (текстовый процессор, настольные издательские системы, электронные таблицы, системы управления базами данных и др.) можно отнести к инструментарию.

К наиболее распространенным относятся следующие информационные технологии:

- обработка текстовой информации;
- обработка числовой информации;
- обработка графической информации;
- базы и банки данных, в том числе учебные;
- пакеты прикладных программ;
- экспертные обучающие системы;
- мультимедиа;
- системы виртуальной реальности, в том числе тренажеры;
- системы искусственного интеллекта;
- телекоммуникационные системы, включая Интернет.

Информационные технологии находят широкое применение в различных сферах человеческой деятельности. Часто они используются в автоматизированных информационных системах.

Информационная система — взаимосвязанная совокупность средств, методов и персонала, участвующих в обработке данных.

Информационные системы бывают разомкнутые и замкнутые.

В разомкнутой информационной системе получаемая потребителем информация используется свободно. В замкнутой информационной системе существует тесная связь информации с ее структурами и потребителем.

Структура любой информационной системы состоит из совокупности обеспечивающих и функциональных подсистем. К обеспечивающим подсистемам относятся:

1) техническое обеспечение;
2) математическое обеспечение;
3) программное обеспечение;
4) информационное обеспечение;
5) правовое, организационное и другое обеспечение.

Наличие функциональных подсистем зависит от целевого назначения системы. В настоящее время широко распространены автоматизированные информациолнные системы.

Примеры реализации автоматизированных систем.

АСУ — автоматизированные системы управления — комплекс технических и программных средств, которые во взаимодействии с человеком организуют управление объектами в производстве или общественной сфере. Например, АСУ-ВУЗ и т. д.

АСУТП — автоматизированные системы управления технологическими процессами. Например, управление работой станка с числовым программным управлением (ЧПУ), процессом запуска космического корабля и т. д.

АСНИ — автоматизированные системы научных исследований — программно-аппаратный комплекс, в котором научные приборы сопряжены с компьютером, вводят в него данные измерений автоматически, а компьютер производит обработку этих данных и представление их в наиболее удобной для исследователя форме.

АОС — автоматизированные обучающие системы. Помогают учащимся осваивать новый материал, производить контроль знаний, преподавателям — готовить учебные материалы и т. п.

САПР — системы автоматического проектирования — программно-аппаратный комплекс, который во взаимодействии с человеком (конструктором, инженером, архитектором и др.) позволяет максимально эффективно проектировать механизмы, здания, узлы сложных агрегатов и др.

Широко распространены диагностические системы в медицине, системы организации продажи билетов, ведения бухгалтерско-финансовой деятельности, редакционно-издательской деятельности и др.

Процесс перехода индустриального общества к информационному обществу начался во 2-й половине XX в. и получил название «информатизация», т. е. процесс создания, развития и всеоб-

щего применения средств и технологий, обеспечивающих достижение и поддержание уровня информированности всех членов общества. Информатизация при этом становится стратегическим ресурсом общества и занимает ключевое место в экономике. Информационное общество должно иметь высокоразвитую информационную среду, которая включает деятельность человека по созданию, переработке, передаче и накоплению информации.

Основные характеристики информационного общества:

1. 80 % работающих занято в сфере производства, хранения, переработки, обмена, продажи информации и информационных услуг.

2. Любому члену общества обеспечивается доступ к необходимой ему информации в соответствии с законодательством.

3. Информация — важнейший стратегический ресурс, занимающий ключевое место в экономике, образовании, культуре, т. е. во всех сферах.

Информационное общество — это общество структуры, техническая база и человеческий потенциал которого приспособлены для оптимального превращения знаний в информационный ресурс и переработки последнего с целью перевода пассивных форм в активные. В информационном обществе большинство работающих людей заняты производством, хранением, переработкой, продажей и обменом информации.

4.2. Этапы развития информационных технологий. Информационные революции

Кратко можно выделить следующие этапы развития информационных технологий:

1. «Ручная» информационная технология (до 2-й половины XIX в.). Инструментарий: перо, чернильница, бухгалтерская книга. Форма передачи информации: почта. Коммуникации осуществляются ручным способом путем почтовой пересылки писем, пакетов. Цель: представить информацию в нужной форме.

2. «Механическая» информационная технология (с конца XIX в.). Инструментарий: пишущая машинка, телефон, фонограф. Передача информации с помощью усовершенствованной почты. Цель: предоставить информацию в нужной форме, но более удобными средствами.

3. «Электрическая» информационная технология (40—60-е годы XX в.). Инструментарий: большие ЭВМ и соответствующее программное обеспечение, электрическая пишущая машинка, портативный магнитофон, копировальные аппараты. Цель: не только предоставление информации в нужной форме и более удобными средствами, а также контроль ее содержательной части (уделяется внимание не только форме, но и содержанию).

4. «Электронная» информационная технология (с начала 1970-х годов). Инструментарий: большие ЭВМ и создаваемые на их базе АСУ, оснащенные широким программным обеспечением. Цель: формирование содержательной части информации.

5. «Компьютерная» («новая») информационная технология (с середины 1980-х годов). Инструментарий: персональный компьютер с большим количеством программных продуктов различного назначения. Система поддержки принятия решений, искусственный интеллект. Реализуется на ПК, использует телекоммуникационную связь. Используются микропроцессоры. Цель: содержание и доступность для широкого потребителя.

Этапы появления средств и методов обработки информации, вызвавших *кардинальные* изменения в обществе, определяются как *информационные революции.*

Первая информационная революция связана с изобретением письменности, обусловившей гигантский качественный и количественный скачок в развитии цивилизации. Появилась возможность хранения и накопления информации и передачи ее последующим поколениям. Появились средства и методы накопления информации.

Вторая информационная революция (середина XVI в.) связана с появлением книгопечатания, изменившего человеческое общество, культуру и организацию деятельности самым радикальным образом. Человек получил удобные средства тиражирования информации. Книгопечатание открыло возможность нового обмена информацией, возможность для каждого человека самостоятельного изучения книг — источника знаний. С его помощью открылся абсолютно новый способ хранения информации.

Третья революция (конец XIX в.) связана с изобретением электричества, благодаря которому появились телеграф, телефон, радио, позволяющие оперативно передавать и накапливать информацию в любом объеме. Значение этой революции в том, что появились средства информационной коммуникации, и передача информации теперь происходит на большие расстояния.

Четвертая революция (70-е годы XX в.) связана с изобретением микропроцессорной технологии и появлением персонального компьютера. Произошел окончательный переход от механических и электрических средств преобразования информации к электронным, что привело к миниатюризации всех приборов и устройств. Стали делать маленькие компьютеры. На микропроцессорах и интегральных схемах создаются компьютеры, компьютерные сети, системы передачи данных. Главная роль этой революции заключается в том, что появились персональные компьютеры и получили развитие автоматизированные информационные технологии.

В связи с переходом на микропроцессорную базу значительно изменяются технические средства бытового, культурного и прочего назначения. В различных областях начинается широкое использование телекоммуникационной связи, компьютерных сетей.

4.3. Информационная культура

Понятие «культура» может быть определено как уровень развития творческих сил и способностей человека. Это понятие определяет знания и умения, профессиональные навыки, уровень интеллектуального, эстетического и нравственного развития, мировоззрение, способы и формы взаимного общения людей. Для жизни и работы в информационном обществе человек должен быть подготовлен к быстрому восприятию и обработке больших объемов информации; ему необходимо овладеть современными средствами, методами и технологией работы. В новых жизненных условиях степень информированности одного человека напрямую зависит от информации, приобретенной другими людьми. Поэтому уже недостаточно уметь самостоятельно осваивать и накапливать информацию, а следует научиться такой технологии работы с информацией, когда решения подготавливаются и принимаются на основе коллективного знания. Таким образом, человек должен иметь определенный уровень культуры для работы с информацией, т. е. должен обладать определенной информационной культурой.

Информационная культура — умение целенаправленно работать с информацией и использовать для ее получения, обработки и передачи компьютерную информационную технологию, современные технические средства и методы.

Информационная культура является продуктом разнообразных творческих способностей человека и важнейшей составляющей культуры в целом. В основном она проявляется в следующем:

- в конкретных навыках по использованию различных технических устройств — от телефона до ПК и компьютерных сетей;
- в способности использовать в своей работе компьютерную информационную технологию;
- в умении извлекать информацию из различных источников — от периодической печати до электронных коммуникаций;
- в умении представлять информацию в понятном виде и эффективно ее использовать;
- в знании аналитических методов обработки информации;
- в умении работать с различными видами информации.

Информационная культура заимствует и использует достижения многих наук: кибернетики, информатики, теории информации, математики, теории проектирования баз данных и ряда других дисциплин. Неотъемлемой частью информационной культуры является знание информационной технологии и умение применять ее на практике. В информационном обществе необходимо овладевать информационной культурой с раннего детства, сначала с помощью электронных игрушек, а позднее — с привлечением ПК. Обеспечить должный уровень информационной культуры призваны в первую очередь учебные заведения, в которых изучаются компьютеры, компьютерные информационные технологии, информационные системы, современные средства и методы обработки информации, системы искусственного интеллекта, компьютерные коммуникации.

Глава 5
ТЕХНОЛОГИИ ОБРАБОТКИ ИНФОРМАЦИИ

5.1. Технологии обработки текстовой информации

Название программных средств, предназначенных для создания, редактирования и форматирования простых и комплексных текстовых документов, — *текстовые процессоры*. В настоящее время в России наибольшее распространение получил текстовый процессор *Microsoft Word*. Это связано, прежде всего, с тем, что создатели относительно давно предусмотрели *локализацию* программы в России путем включения в нее средств поддержки работы с документами, исполненных на русском языке.

Основные версии текстового процессора Microsoft Word. Первые версии текстового процессора Microsoft Word относятся к 1980-м годам и, соответственно, к операционной системе MS-DOS. Последней версией процессора для неграфической ОС была версия *Microsoft Word 5.0*. Она позволяла создавать, редактировать и распечатывать форматированные текстовые документы. Но она не могла соблюдать принятый ныне принцип соответствия экранного изображения печатному (принцип WYSIWYG), и операции форматирования выполнялись в известной степени «вслепую». Однако возможность просмотра документа в «натуральном» виде все-таки была. Она реализовывалась специальным режимом *предварительного просмотра* (*Preview*). Основным преимуществом текстового процессора Word 5.0 была возможность встраивания в текст графических объектов, правда, без обтекания графики текстом.

Принцип WYSIWYG впервые был реализован в версии *MS Word for Windows* (*Word 6.0*). Благодаря этому принципу значительно упростились и стали наглядными приемы форматирова-

ния документов. Будучи приложением *Windows 3.1*, программа могла использовать системный буфер обмена, а пользователи получили мощное средство для создания комплексных документов.

Следующая версия программы, *Microsoft Word 95* (*Word 7.0*), ориентирована на графическую ОС *Windows 95*. После этой версии текстовый процессор уже не рассматривается только как отдельное приложение. В состав мощного офисного пакета Microsoft Office входит несколько приложений (с каждой новой версией этот состав расширяется), и на процессор Microsoft Word возлагаются дополнительные функции интеграции прочих приложений. Он позволяет организовать эффективный обмен данными между составляющими приложениями. Еще одним важным нововведением седьмой версии стало управление взаимодействием текста со встроенными объектами.

Особенный успех этой версии в России имели встроенные средства поддержки русского языка (автоматическая проверка орфографии и грамматики).

Восьмая версия программы, *Microsoft Word 97* (*Word 8.0*), вошедшая в *Microsoft Office 97*, внесла относительно мало изменений в повседневную офисную работу. Например, ее жесткая ориентация на использование шрифтов *Unicode* затруднила обмен данными с большинством приложений, выпущенных сторонними фирмами. Возможность сохранения документов в электронных форматах *HTML* и *PDF*, рассчитанная на публикацию документов в Интернете, осталась непроработанной и не вошла и практику веб-дизайнеров. Начиная с этой версии, процессор Microsoft Word можно рассматривать как средство автоматизации авторской деятельности при создании электронных и печатных документов.

Для создания электронных и печатных документов используют разные приемы и методы. Применение неадекватных средств значительно усложняет этапы работы с документом.

В версии текстового процессора *Microsoft Word 2000* (*Word 9.0*) пакета *Microsoft Office 2000* устранены основные недостатки предыдущей версии, заметно улучшена система управления и введены мощные средства поддержки сетевых режимов работы.

Версия текстового *процессора Word XP* (*Word 10.0*) входит в состав пакета *Microsoft Office XP*. В ней заметно расширены средства работы со стилями и шаблонами, введены механизмы, позволяющие автоматически обеспечить единство оформления документа.

Версия текстового *процессора Word 2007* входит в состав пакета *Microsoft Office 2007* и работает под управлением *Windows Vista*. Имеет новый интерфейс, новые возможности безопасности и поддержки.

Рабочее окно процессора Microsoft Word XP. Основные элементы управления рабочего окна процессора Microsoft Word XP: строка меню, панель инструментов, рабочее поле и строка состояния, включающая индикаторы.

Типы документов и режимы их отображения. В MS Word можно составлять документы трех типов: печатные (для распечатки на принтере), электронные (передаются заказчику в виде файлов, веб-документы (для использования в сети Интернет). Документы создаются с помощью команд **Файл → Создать → Создание**, и далее выбирается режим создания документа.

Режимы отображения:

- *обычный*, когда представляется только содержательная часть документа без реквизитных элементов оформления, относящихся не к тексту, а к печатным страницам (колонтитулы, колонцифры, сноски и др.). Этот режим используют в случаях, когда содержательная часть документа имеет большее значение, чем внешнее представление;
- *веб-документа*, когда экранное представление не совпадает с печатным. Понятие печатной страницы для электронных документов не имеет смысла, поэтому назначенные параметры страницы не учитываются, а форматирование документа на экране является относительным. В этом режиме разрабатывают электронные публикации;
- *разметки*, когда экранное представление документа полностью совпадает с печатным, вплоть до назначенных параметров печатной страницы. Используют для печати;
- *структуры*, когда можно отобразить только заголовки документа. Режим полезен, когда разработку начинают с создания плана содержания документа с помощью панели инструментов **Структура**;
- *схемы документа*, когда окно приложения имеет две рабочие панели. На левой панели представляется структура документа, а на правой — сам документ.

Выбор режимов осуществляют с помощью команд меню **Вид**. Режимы предварительного просмотра осуществляют с помощью команд **Файл → Предварительный просмотр**.

Панели инструментов — 18 панелей (**Вид → Панели инструментов**): **Стандартная**, **Форматирование**, **Visual Basic**, **WordArt**, **Автотекст**, **Базы данных**, **Веб-компоненты**, **Веб-узел**, **Настройка изображения**, **Рамки**, **Рецензирование**, **Рисование**, **Слияние**, **Статистика**, **Структура**, **Таблицы и границы**, **Формы**, **Элементы управления**.

Панель **Область задач**: **Создание документа**, **Буфер обмена**, **Поиск**, **Вставка картинки**, **Стили и форматирование**, **Показать форматирование**, **Слияние** (**Сервис → Письма и рассылки → Мастер слияния**), **Перевод** (**Сервис → Язык → Перевод**).

Приемы работы с текстами в процессоре. К базовым приемам работы с текстами в текстовом процессоре относятся следующие:

- создание документа;
- ввод текста;
- редактирование текста;
- рецензирование текста;
- форматирование текста;
- сохранение документа;
- печать документа.

Принято использовать два метода создания нового документа: на основе готового шаблона или на основе существующего документа. Второй метод потенциально опасен и категорически не рекомендуется! Тем не менее им пользуются часто. Он применяется начинающими пользователями, не умеющими создавать шаблоны и пользоваться ими. Шаблоны — это те же образцы документов, но защищенные от досадных неприятностей. Создание документа на основе шаблона выполняется следующим образом. Команды меню **Файл → Создать** открывает панель **Область задач** в режиме создания документа. Щелкните на этой панели по ссылке **Общие шаблоны** — откроется диалоговое окно **Шаблоны**. Надо включить переключатель **Создать документ** и выбрать подходящий шаблон. Если никаких предпочтений нет, следует выбрать шаблон **Новый документ** на вкладке **Общие**. Созданный документ приобретает имя Документ1, принятое по умолчанию. Его целесообразно сразу же сохранить под «правильным» именем, выбрав для него соответствующую папку и выполнив команды меню **Файл → Сохранить как**.

Режимы вставки и замены символов. Текст вводится в режимах вставки и замены. Для переключения режимов следует два раза щелкнуть мышью в строке состояния по индикатору **Зам**. На клавишу **Enter** нажимают для завершения текущего абзаца и

перехода к следующему. Конец абзаца помечается в документе с помощью непечатаемого символа ¶ (пи) — маркера абзаца. Объединение двух абзацев означает удаление символа конца абзаца между ними. Разбиение одного абзаца на два достигается вставкой символа в место разбиения текста. Для того чтобы внутри абзаца перейти на новую строку, следует нажать сочетание клавиш **Shift+Enter**.

Ввод специальных и произвольных символов. Основным средством для их ввода служит окно **Символ** (**Вставка** → **Символ**). Данное диалоговое окно имеет две вкладки: **Символы** и **Специальные знаки**.

Средства автоматизации. В процессоре имеются средства, позволяющие автоматизировать ввод текста. **Автотекст** — это режим автоматического ввода фрагментов текста. Он представлен двумя функциями: автозавершением и автотекстом. Процессор хранит словарь. При вводе первых четырех символов словарного элемента на экране появляется всплывающая подсказка с полным текстом слова или фразы. Завершает пользователь ввод всего фрагмента нажатием клавиши **Enter** — *автозавершением*. Однако пользователь может самостоятельно выбрать необходимый элемент текста из списка с иерархической структурой — это функция *автотекста*. Список элементов автотекста открывается с помощью панели инструментов **Автотекст** (**Вид** → **Панели инструментов** → **Автотекст**).

Настройку словаря автотекста выполняют в диалоговом окне **Автозамена** (**Вставка** → **Автотекст** → **Автотекст**). Простейший способ наполнения словаря новым содержимым — выделить текст на экране, щелкнуть по кнопке **Автотекст** и в открывшемся диалоговом окне использовать кнопку **Добавить**.

Использование средства Автозамена при вводе. Длинные последовательности символов можно заменить произвольным сочетанием других символов.

Настройку средства **Автозамена** выполняют в одноименном диалоговом окне (**Сервис** → **Параметры автозамены**). Для этого надо установить флажок **Заменять при вводе**, ввести заменяемую комбинацию в поле **Заменить**, а замещающую комбинацию — в поле **На**, после чего пополнить список автозамены щелчком по кнопке **Добавить**.

Средства автоматизации проверки правописания. Это средства проверки орфографии и грамматики. Реализовано два режима

проверки: автоматический и командный (флажки **Автоматически проверять орфографию** и **Автоматически проверять грамматику** на вкладке **Правописание** диалогового окна **Параметры** (**Сервис** → **Параметры** → **Правописание**)).

Рецензирование текста. Это редактирование текста с регистрацией изменений и комментирование текста (**Вид** → **Панели инструментов** → **Рецензирование**). С помощью панели **Рецензирование** можно:

- создавать, просматривать и удалять примечания;
- регистрировать, просматривать, принимать и отменять изменения;
- выбирать цвета выделения примечаний;
- сохранять версии документа.

Форматирование текста. Виды форматирования: символов, абзацев, документа в целом.

Основные приемы:

- выбор и изменение гарнитуры шрифта;
- управление размером шрифта;
- управление начертанием и цветом шрифта;
- управление методом выравнивания;
- создание маркированных и нумерованных списков;
- управление параметрами абзаца.

Настройку шрифта выполняют в диалоговом окне **Шрифт** (**Формат** → **Шрифт**). В версии Word XP данное диалоговое окно имеет три вкладки: **Шрифт**, **Интервал** и **Анимация**.

На вкладке **Шрифт** выбирают:

- гарнитуру шрифта;
- размер шрифта (измеряется в полиграфических пунктах);
- вариант начертания;
- цвет символов;
- наличие подчеркивания;
- характер видоизменения.

Для электронных документов, которые читают с экрана, многие предпочитают применять рубленые шрифты.

При форматировании документа в целом можно выполнять:

- установку параметров страницы (поля, размер бумаги, ориентацию листа);
- разбивку на страницы и разделы;
- вставку номеров страниц, колонтитулов, сносок, закладок, примечаний и др.;

- создание (или вставку) формул, рисунков, таблиц, видео- или звуковых фрагментов;
- вставку названий, иллюстраций, перекрестных ссылок;
- форматирование оглавления, указателей, списков, иллюстраций и др.

Можно использовать следующие команды (рис. 5.1).

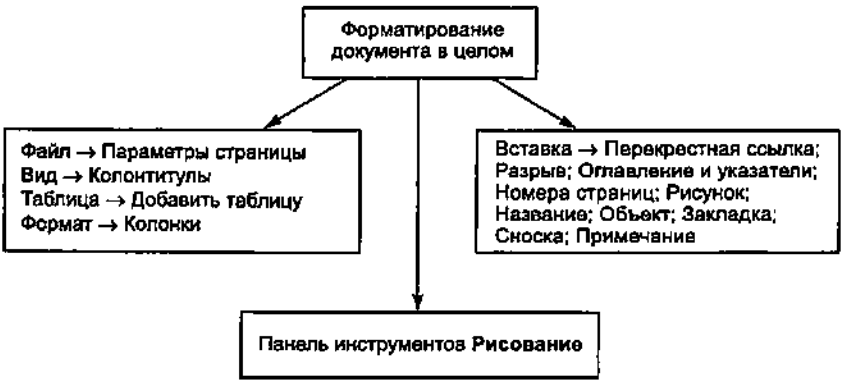


Рис. 5.1. Команды для форматирования документа

Поддерживается *четыре типа выравнивания*: по левому краю, по центру, по правому краю, по ширине.

Команды создания маркированных, нумерованных и многоуровневых списков: **Формат → Список**.

Наиболее общими средствами автоматизации оформления документов являются стили оформления абзацев, шаблоны документов, темы оформления.

Стиль оформления — именованная совокупность настроек параметров шрифта, абзаца, языка и некоторых элементов оформления абзацев (линий и рамок). Благодаря использованию стилей обеспечивается простота форматирования абзацев и заголовков текста, а также единство их оформления в рамках всего документа. Настройка стиля: **Формат → Стиль**.

Шаблоны — заготовки будущих документов. Все документы Word создаются на основе выбранного шаблона. Шаблон — служебный файл, который содержит информацию о структуре и оформлении документов определенного типа. Шаблоны являются специализированными программами и сохраняются как файлы с расширением .dot. При запуске Word без указания имени

файла по умолчанию создается документ под условным наименованием Документ 1 на основе стандартного шаблона **Обычный**.

Темы — совокупность следующих элементов оформления:

- фоновый узор;
- стили оформления основного текста и заголовков;
- стиль оформления маркированных списков;
- стиль графических элементов оформления (линий).

Выбор темы: **Формат → Темы**.

В Word можно создавать сложные и составные документы.

Сложные документы содержат информацию разного типа (текст, таблицы, диаграммы, формулы, рисунки и др.). *Составные документы* включают главный и вложенные документы. Рекомендации по созданию документов обоих типов даются в разделах 5.1.1 и 5.1.2. Здесь только кратко рассматривается создание в Word некоторых компонентов сложных документов.

Создание таблиц. Таблица в программе Word может содержать максимум 31 столбец и произвольное число строк. На пересечении столбцов и строк находятся ячейки, в которых размещают данные: текст, числа, графику и др. Ячейки таблицы имеют адреса, образованные из имени столбца и номера строки (табл. 5.1).

Таблица 5.1. **Пример таблицы, созданной в программе Word**

	A	B	C
1	A1		
2			C2

Имеется несколько способов создания таблиц:

- с помощью команд меню **Таблица → Нарисовать таблицу...**;
- с помощью кнопки **Добавить таблицу....**

Можно вставить лист таблицы MS Excel, выбрав команды меню **Вставка → Объект → Лист MS Excel**.

Ранее созданный текст можно преобразовать в таблицу: **Таблица → Преобразовать → Текст в таблицу....** Но перед этим текст следует подготовить с помощью клавиш **Enter**, **Tab** и пробелов.

Для форматирования таблиц используют все команды меню **Формат**, контекстного меню и кнопки **Таблицы и границы**.

Заголовки, как правило, создают в первой строке таблицы: **Таблица → Заголовки**.

Можно выполнять несложные вычисления в таблицах программы Word. Например, сумму чисел в строке или столбце можно вычислить с помощью кнопки **Автосумма**, предварительно выделив слагаемые значения. При вычислениях по формуле следует установить курсор в ячейку результата, выполнить команды меню **Таблица → Формула**, ввести данные в диалоговом окне **Формула** или выбрать нужную функцию. Формула вводится после знака =, адреса ячеек задаются в латинском регистре.

Адреса ячеек разделяются запятыми, а диапазон ячеек — двоеточием A1:B4. Ссылка на всю строку — 1:1 (первая строка). Ссылка на весь столбец — A:A (столбец A).

Знаки операций: +, –, =, *, /, %, ^ и др.

Ключевые слова: LEFT, RIGHT, ABOVE (выше), BELOW (ниже).

Разделитель: ; (точка с запятой).

Встроенные функции:

- SUM, например SUM A1:E6 или SUM(ABOVE);
- PRODUCT (произведение), например PRODUCT(A1;E6);
- MAX, MIN;
- COUNT (подсчет числа значений в диапазоне ячеек), например COUNT(A1:B6; B10);
- AVERAGE (вычисление среднего значения для диапазона ячеек), например AVERAGE(A1:E6; B10). Результат вычисления вставляется в виде поля.

Создание диаграмм. Диаграммы (рис. 5.2) можно создавать следующим образом:

- с помощью встроенного в Word редактора диаграмм MS Graph (команды: **Вставка → Объект → Создание → Диаграмма MS Graph**);

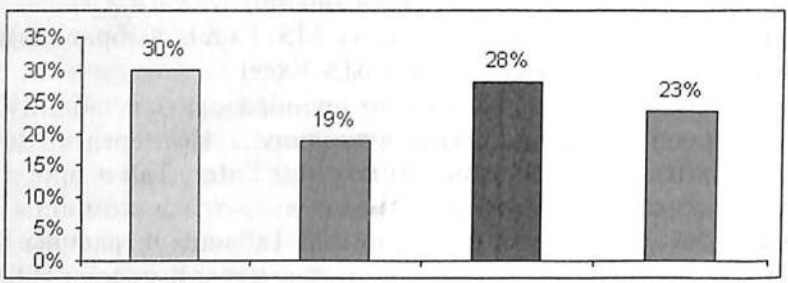


Рис. 5.2. Пример диаграммы

- путем вставки готовых диаграмм из других документов (кнопка **Вставить диаграмму**).

Создание математических формул. Математические формулы можно создавать с помощью встроенного редактора формул MS Equation. В документе следует установить курсор в место вставки формулы и для вызова редактора выбрать команды **Вставка → Объект → Создание → MS Equation 3.0**. Появится панель со значками математических формул, которые нужно использовать для создания формулы (рис. 5.3).

Для редактирования формулы следует использовать команды меню окна редактора формул: **Правка → Объект → Equation → Открыть**.

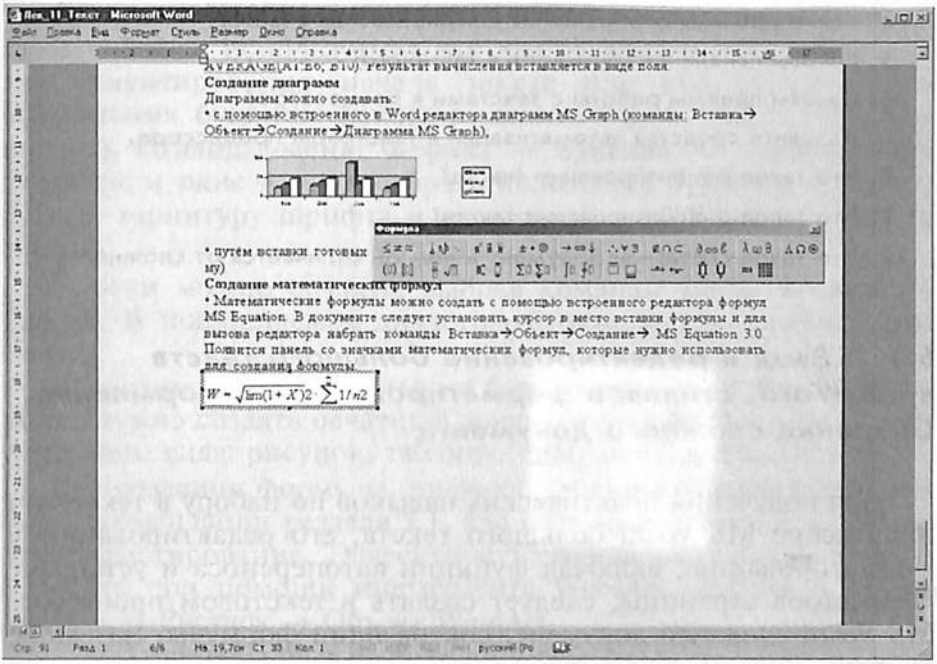


Рис. 5.3. Пример создания математической формулы в документе Word

Создание рисунков. Рисунки можно создавать:

- с помощью панели **Рисование**;
- с помощью команд **Вставка → Объект → Точечный рисунок** выйти в редактор Paint и, создав там рисунок, скопировать его в документ Word;

- вставлять из других файлов с помощью команд меню **Вставка → Рисунок...**.

Привязку рисунка к тексту можно осуществить, используя команды **Формат → Рисунок → Положение**.

Для слияния документов следует использовать панель **Слияние**.

Контрольные вопросы

1. Что такое текстовый процессор?
2. Каковы версии текстового процессора Microsoft Word?
3. Какие типы документов можно создавать в MS Word?
4. Какие режимы отображения документов вы знаете?
5. Каковы панели инструментов в MS Word?
6. Каковы приемы работы с текстами в процессоре?
7. Назовите средства автоматизации в текстовом процессоре.
8. Что такое рецензирование текста?
9. Что такое форматирование текста?
10. Что такое составной документ и чем он отличается от сложного?

5.1.1. Ввод и редактирование большого текста в MS Word, стилевое форматирование и оформление. Создание сложного документа

Для получения практических навыков по набору в текстовом процессоре MS Word большого текста, его редактированию и форматированию, включая функции автопереноса и установки параметров страницы, следует создать в текстовом процессоре MS Word сложный документ, содержащий формулы, названия, буквицу, сноски, иллюстрации, таблицы, диаграммы.

Установка параметров страниц (полей, размера бумаги, источника бумаги и др.) производится выбором в меню **Файл** пункта **Параметры страницы**. Если предусматривается переплет документа, то при установке размера полей необходимо увеличить размер внутреннего (левого) поля до величины переплета или установить величину переплета отдельно. Следует указать расстояние от краев страницы до колонтитулов.

Для того чтобы избежать появления жидких строк (строк с большими межсловными интервалами), следует включать функцию автопереноса.

Функцию автопереноса следует установить, выбрав команды меню **Сервис → Язык → Расстановка переносов → Автоматическая расстановка переносов**.

При формировании оглавления следует использовать стилевое форматирование. Каждый выбранный заголовок для оглавления форматировать стилем (например, **Заголовок 1**) с помощью команд **Формат → Стиль → Заголовок 1 → Применить**. Затем курсор установим в то место, где должно быть оглавление. Используем команды меню **Вставка → Оглавление и указатели → Оглавление**. Будет создано оглавление к документу.

Буквица — это увеличенная в размере первая буква первой строки абзаца, используемая как для оформления текста, так и для акцентирования начала текста или его раздела. Для оформления буквицы нужно выделить букву, после чего выполнить команды меню **Формат → Буквица**. В появившемся диалоговом окне нужно выбрать положение буквицы в тексте, размер, гарнитуру шрифта и расстояние от буквицы до текста абзаца.

Сноски можно создать, выбрав команды меню **Вставка → Сноска**. В появившемся диалоговом окне нужно выбрать вид сноски.

Допустим, например, что на базе имеющегося рукописного текста нужно создать печатный документ, включив в него объекты разного вида: рисунок, таблицу, диаграмму, формулу и др.

Для создания формулы, рисунка, таблицы следует использовать рекомендации раздела 5.1, для создания названий — стилевое форматирование. Эффектное название (заголовок) можно получить при помощи инструмента **WordArt**, выбрав в меню: **Вставка → Рисунок → Объект WordArt**.

Последовательность выполнения работы:

1. Создаем в MS Word документ и сохраняем его под именем Prozed.doc.

2. Открываем документ и вводим в него (набирая на клавиатуре) исходный текст.

3. Выполняем редактирование и форматирование текста, установив параметры страницы и функцию автопереноса. Для стилевого форматирования заголовка применяем стиль **Заголовок 2**.

4. Формируем оглавление, используя стилевое фрматирование.
5. Открываем текстовый документ Prozed.doc и создаем из него сложный документ, вводя рисунок, таблицу, диаграмму, формулу, форматируя заголовки.
6. Создаем буквицу и сноску.
7. Формируем оглавление.

Пример созданного документа

Текст

Программы-процедуры[1] — программы, которые не возвращают ни одного значения или возвращают более одного значения. Для обозначения таких программ в Паскале используют служебное слово `procedure`, а общий вид следующий:

```
Procedure имя процедуры (список_аргументов);
```

Описание внутренних переменных программы:

```
Begin
     Тело программы
End;
```

Имя процедуры не может использоваться внутри процедуры как псевдопеременная.

Для существования треугольника необходимо, чтобы Σ любых его сторон была не меньше третьей.

Пример: Вычисление площади треугольника с помощью процедур.

```
Program Prim {подпрограммы процедуры}
Var
X1, X2, X3: integer; Procedure ptreug (a, b, c: real;
var stren: real; var est: bolean);
Var
P:real;
Begin
If(a+b<c) or (a+c<b) or (c+b<a) then
Est:=False
```

[1] Понятие ввел Уилкс в 1957 г.

```
Else
Begin
East:=true;
P:=(a+b+c)/2;
Stren:=sqrt((p*(p-a)*(p-b)*(p-c))
End;
End.
```

Этот указатель поддерживается файловой системой и прикладному программисту недоступен. В языках программирования есть. Специальные процедуры изменения указателя текущей записи в файле. Значения указателя текущей записи задается величиной длинного целого типа (`long`) процедура установки текущего указателя в Паскале обозначается `SEEK`. Например,

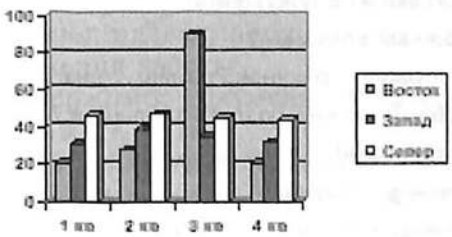
```
SEEK(fb, POS-1); {переход от POS к POS-1, чтобы отчёт шёл
от 0}
```

возникли, чтобы сохранить для дальнейшего использования изображения текстов на экране. Текстовые файлы являются аналогом текстового изменения на экране, создаваемого операторами ввода.

Таблица наблюдений

Группы	1 кв. (ч)	2 кв. (ч)	3 кв. (ч)	4 кв. (ч)
Восток	20	29	90	20
Запад	27	39	33	30
Север	45	45	43	41

Диаграмма



Рисунок

Формула

$$\sqrt{D} = \sqrt{b^2 - 4ac}$$

Оглавление

1. Текст *256*
2. Таблица наблюдений *257*
3. Диаграмма *257*
4. Рисунок *258*
5. Формула *258*

Контрольные вопросы

1. Что понимается под параметрами страницы и как установить их?
2. Какие соотношения между размерами полей необходимо соблюдать и почему?
3. Что такое жидкие строки и как задать функцию автопереноса?
4. Как создать оглавление документа?
5. Что такое сложный документ?
6. Как создать формулу, рисунок, таблицу, диаграмму?
7. Какими способами можно создать таблицу в MS Word?
8. Как создать заголовки?
9. Каково назначение WordArt?
10. Что такое буквица и как ее оформить?

5.1.2. Создание составного документа (главного и вложенных). Макетирование страниц. Оформление титульного листа

Составной документ содержит главный документ и вложенные документы.

Для создания главного документа надо сначала выделить текст, а затем выполнить команды **Вид → Структура**. Появляется специальная панель со стрелками, цифрами и пиктограммами. На ней выбрать кнопку **Главный документ**.

Чтобы создать вложенные документы, следует сначала создать заголовки. Для этого нужно выделить текст и выбрать **Заголовок**. Далее выполнить команды **Вид → Структура** и в специальном меню после выделения объекта выбрать **Создать вложенный документ**.

Вернуться к команде **Разметка страницы** и сделать оглавление, выбрав команды меню **Вставка → Ссылка → Оглавление и указатели**. Установить нужные параметры. Будет создано оглавление со ссылками на вложенные документы. Теперь с его помощью можно открыть любой требуемый документ.

Верстка страницы — это расположение на ней в определенном порядке текста (фрагментов текста), заголовков, иллюстраций, колонтитулов и других элементов оформления.

Многоколоночная верстка — расположение текста в несколько колонок. Текст может занимать всю ширину полосы или располагаться в несколько колонок. Следует выбрать команды **Формат → Колонки**. В открывшемся диалоговом окне нужно выбрать тип колонок, их число, ширину и размер промежутка между ними. Количество колонок определяет их ширину. Чем колонок больше, тем мельче размер шрифта, а это затрудняет восприятие текста.

Титульный лист — лицо документа. На титульном листе следует поместить следующие данные: фамилию и инициалы автора, название документа, место и год издания, элементы художественного оформления: иллюстрации, декоративные элементы.

Порядок выполнения работы:

1. Копируем содержимое документа из примера работы, приведенного в разделе 5.1.1.

2. Части текста, а также таблицу, диаграмму, рисунок и формулу выделяем (заносим) в отдельные документы. Они будут вкладываться в главный документ, где оставлен обычный текст.

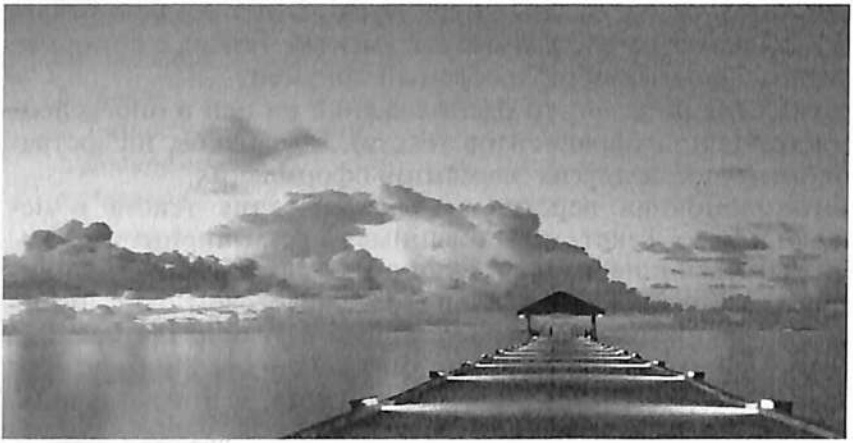
3. Создаем составной документ.
4. Выполняем верстку текста в 1 и 2 колонки.
5. Разрабатываем титульный лист документа.
6. Формируем оглавление.
7. Распечатываем документ.

Пример выполнения работы приведен на рис. 5.4 *а—в*.

МГАВТ

Создание составного документа (главного и вложенных). Макетирование страниц. Оформление титульного листа

(ОТЧЕТ)



Выполнил студент группы СМЭ-4
Иванов Петр

Москва, 2010

Рис. 5.4, *а*. Создание составного документа (титульный лист)

Главный документ

Обычный текст

— возникли, чтобы сохранить для дальнейшего использования изображения текстов на экране. Текстовые файлы являются аналогом текстового изменения на экране, создаваемого операторами ввода. Текстовые строки завершают невидимые служебные символы CR (Carriage Return — возврат каретки) и LF (Line Leed — перевод строки). Это применимо для ОС в IBM PC с использованием клавишы **Enter**.
Пример:

```
Program Primt     {Вывод в текстовый файл}
var
Fa:text;
k, na:integer;
begin
assign (fa, 'primerc.ftx');
rewrite (fa);
for k:=1 to 12 do
writeln (fa, 'квадрат числа', k, 'равен', k*k, 'а-куб', k*k*k);
close (fa)
end.
```

Оглавление (Ссылки на вложенные документы)*

D:\Documenti\Мои документы\Лаб14\Верстка в 2 колонки.doc
D:\Documenti\Мои документы\Лаб14\Таблица.doc
D:\Documenti\Мои документы\Лаб14\\Диаграмма.doc
D:\Documenti\Мои документы\Лаб14\Рисунок.doc
D:\Documenti\Мои документы\Лаб14\Формулы.doc

* Каждый вложенный документ показан не как отдельный, хотя в действительности каждый из них вызывается только по ссылке.

Рис. 5.4, *б.* Создание составного документа (главный документ)

Вложенный документ 1

Верстка текста в 2 колонки

Программы-процедуры — программы, которые не возвращают ни одного значения или возвращают более одного значения. Для обозначения таких программ в Паскале используют служебное слово `Procedure`, а общий вид: `Procedure имя процедуры (список_аргументов);`
Описание внутренних переменных программы:

```
Begin
    Тело программы
End;
```

Имя процедуры не может использоваться внутри процедуры как псевдопеременная. Для существования треугольника необходимо, чтобы Σ любых его сторон была не меньше третьей.

Пример: Вычисление площади треугольника с помощью процедур.

```
Program Prim
Var
X1, X2, X3: integer;
Procedure ptreug (a, b, c: real;
var streu: real; var est: bolean);
```

Формальные аргументы

```
Var
P:real;
Begin
If(a+b<c) or (a+c<b) or (c+b<a) then
Est:=False
Else
Begin
East:=true;
P:=(a+b+c)/2;
Streu:=sqrt((p*(p-a)*(p-b)*(p-c))
End;
End.
Var {здесь задаются переменные
только для главной части программы}
S1, S2: real;
Is: Boolean;
Begin
Ptreug (10, 10, 10, S1, is);
{процедуры названы именем Ptreug}
Writeln ('Площадь треугольника со
сторонами, равными 10, равна
S1:6:3');
---------- ('Введите длины сторон
треугольника);
read (X1, X2, X3);
ptreug (X1, X2, X3, S2, is);
if is then writeln ('площадь
треугольника с этими сторонами
равна' S2:7:2);
else writeln (' треугольника с
такими сторонами не существует")
end.
```

Вложенный документ 2

Таблица (см. таблицу в разделе 5.1.1).

Вложенные документы 3, 4, 5 содержат соответственно диаграмму, рисунок и формулу, которые даны в разделе 5.1.1.

Рис. 5.4, *в*. Создание составного документа (вложенные документы)

Контрольные вопросы

1. Что такое страница-шаблон?
2. Как создать главный и вложенные документы?
3. Чем отличается сложный документ от составного документа?
4. Что значит сверстать страницу?
5. Какова технология многоколончатой верстки?
6. Как создать титульный лист?

5.2. Электронные таблицы Excel. Работа с данными и расчеты в Excel

Для представления данных в удобном виде используют таблицы. Компьютер позволяет представлять их в электронной форме, а это дает возможность не только отображать, но и обрабатывать данные. Класс программ, используемых для этой цели, называется электронными таблицами. Электронная таблица — компьютерный эквивалент обычной таблицы. Табличный процессор — комплекс программ, предназначенный для создания и обработки таблиц.

Электронные таблицы Excel — это самая распространенная и мощная технология для работы с данными. В ячейках таблицы могут содержаться числа, текст, даты, формулы, функции. Главное достоинство электронных таблиц — возможность мгновенного автоматического пересчета всех данных, связанных формульными зависимостями, при изменении значения любого компонента таблицы. В Excel возможности вычисления объединены с богатым набором функций, присущих текстовому, графическому редакторам и другим приложениям пакета MS Office.

Табличный процессор Excel позволяет:

- решать математические задачи: выполнять табличные вычисления, вычислять значения и исследовать функции, строить графики функций, решать уравнения, работать с матрицами и комплексными числами;
- осуществлять математическое моделирование и численное экспериментирование;
- проводить статистический анализ, осуществлять прогнозирование и оптимизацию;

- реализовывать функции базы данных — ввод, поиск, сортировку, фильтрацию и анализ данных;
- вводить пароли либо устанавливать защиту некоторых ячеек или всех ячеек, скрывать фрагменты таблицы или всю таблицу;
- наглядно представлять данные в виде диаграмм и графиков;
- вводить и редактировать тексты, создавать рисунки;
- импортировать, экспортировать и обмениваться данными с другими программами (например, вставлять текст, рисунки, таблицы, приготовленные в других приложениях);
- осуществлять многотабличные связи;
- подготавливать выступления, доклады и презентации.

Версии Excel: Excel 5.0 для Windows 3.11, Excel 7.0 для Windows 95, Excel 97 для Windows 98, Excel 2000 для Windows 2000, Excel для Windows XP и Windows Vista.

Excel находится в файле ...VProgram FilesVMS Office\ excel.exe. Все файлы, созданные программой Excel, имеют расширение .xls. При запуске Excel без указания имени файла процессор по умолчанию предлагает начать создание нового документа под названием «Книга 1». Меню программы Excel (рис. 5.5) отличается от Word тем, что вместо вкладки **Таблица**

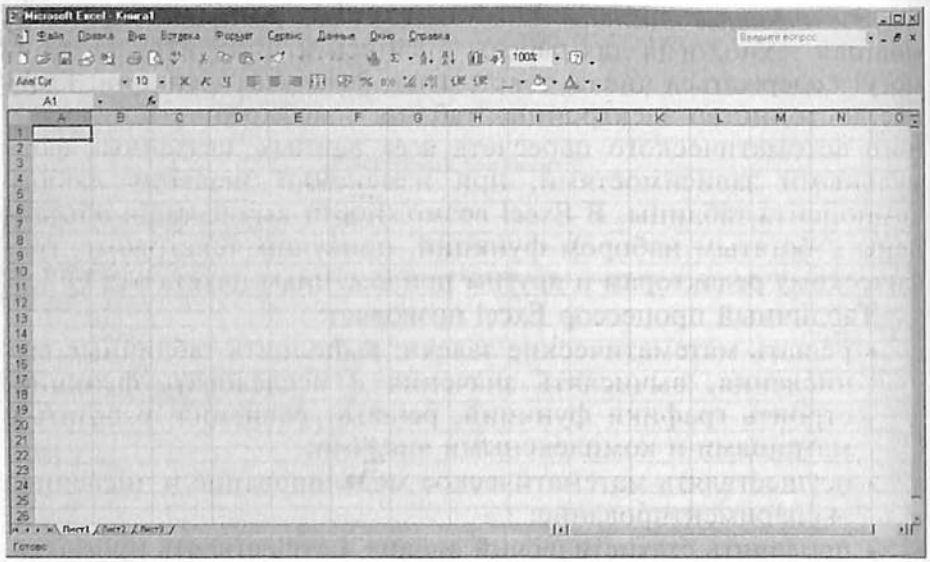


Рис. 5.5. Окно рабочего листа Excel

появляется вкладка **Данные**. На панели инструментов имеются специальные кнопки для форматирования числовых данных: денежный, процентный, разделитель тысяч, уменьшение или разделение разрядности числа. Под панелью инструментов идет строка **Функция**. Слева раскрывается список. В поле имени указывается адрес активной ячейки. Кнопки **XV** = используются для управления процессом ввода и редактирования. На пересечении столбца и строки с обозначением столбцов находится кнопка без надписи для выделения всей таблицы. Затем идет рабочее поле. Ниже него располагается строка с ярлыком рабочих листов. Excel — многооконная программа, т. е. можно одновременно открывать несколько документов. В строке состояния высвечивается режим работы, дополнительная информация и находится поле для автовычисления. Окно можно разделить на 2 или 4 части и одновременно работать с разными частями одной таблицы. Для этого нужно использовать команды **Окно → Разделить**. Чтобы снять разделение: **Окно → Снять разделение**.

Основные понятия Excel. *Рабочая книга* — это совокупность рабочих листов, сохраняемых на диске в одном файле. В каждом файле может размещаться одна книга, а в книге от 1 до 255 рабочих листов. По умолчанию в каждой книге 3 листа. Рабочий лист имеет табличную структуру и может состоять из любого числа страниц, ярлыки активного листа выделяются цветом, а надпись — полужирным текстом. Электронная таблица состоит из 65 536 строк и 256 столбцов. Строки нумеруются числами, а столбцы латинскими буквами. Заголовок столбца или строки служит не только для обозначения, но и для изменения.

Ячейка — это область электронной таблицы, находящаяся на пересечении столбца и строки, это наименьшая структурная единица на рабочем листе. Формат и размеры ячеек можно изменить с помощью мыши или команд меню. Каждая ячейка имеет адрес, который используется при ссылке на ячейку, например A1, AB2.

Ссылка — это способ указания адреса. Адрес и содержимое текущей ячейки выводятся в строке **Функция**. Адреса ячеек могут быть относительными и абсолютными, ячейки могут иметь собственные имена. Блок может состоять из одной ячейки, строки или ее части, столбца или его части, а также последовательности строк и столбцов. Блок ячеек может быть выделен с помощью мыши, а также непосредственным набором с клавиатуры на-

чального и конечного адресов. Для отмены выделения есть команда **Разгруппировать листы**.

Обозначение ячейки, составленное из заголовка столбца и номера строки, называется *относительным адресом*. Ссылки на диапазон ячеек состоят из адреса ячейки в левом верхнем углу, двоеточия и адреса ячейки в правом нижнем углу, например A1:C5. Для обозначения адреса ячейки с указанием адреса листа необходимо ввести, например, такой текст: «Лист 2! И5» или «[Книга 1] Лист 2! B5». При копировании формул в Excel действует правило *относительной адресации ячеек*. Табличный процессор автоматически смещает адрес (в соответствии с относительным расположением исходной ячейки). Если же ссылка на ячейку не должна изменяться при копировании, то вводят абсолютный адрес ячейки.

Абсолютная ссылка создается из относительной путем вставки знака доллара перед заголовком столбца и номером строки ($A$4).

Смешанный адрес — это адрес, постоянным в котором является только один из компонентов ($A1, A$1). Изменение типа ссылки для редактирования адреса производится с помощью клавиши **F4**. Удобный способ ссылки на адрес ячейки путем присвоения ей произвольного имени. Имена используют в формулах вместо адресов.

Имена — это абсолютные ссылки. Собственными именами можно обозначить постоянные величины, коэффициенты и др. Присвоить имя ячейке можно с помощью команд **Вставка → Имя → Присвоить** или используя поле имени либо сочетание клавиш **Ctrl+F3**. Если требуется присвоить имя из текста, то нужно выполнить команды **Вставка → Имя → Создать → Используется из текста**. Для быстрого перехода к ячейкам, которым присвоены имена, используют клавишу **F5** и вводят нужное имя в поле имен в раскрывшемся диалоговом окне. Для быстрой вставки имени в формулу используют клавишу **F3** и диалоговое окно **Вставить имя**.

Правила: имя не может начинаться с цифры и не должно быть похоже на адрес ячейки, нельзя использовать пробелы — только символ подчеркивания (Доходы_за_2004), длина имени должна быть не более 255 символов. Имена листов должны содержать не более 31 символа, их нельзя заключать в квадратные скобки, нельзя использовать знаки :, /, \, ?, *. Переименовывать листы можно с помощью меню **Правка**.

Типы данных: текст, числа, даты, функции, формулы.

Функции — это программы с уникальными именами, для которых пользователь должен задать аргументы. Все функции имеют одинаковый формат записи:

ИМЯ ФУНКЦИИ (ПЕРЕЧЕНЬ АРГУМЕНТОВ).

Для облегчения работы со вложенными функциями используется Мастер функций (рис. 5.6), который имеет около 400 встроенных функций. Например, математические, тригонометрические, статистические, инженерные, даты и времени, финансовые, информационные, логические функции.

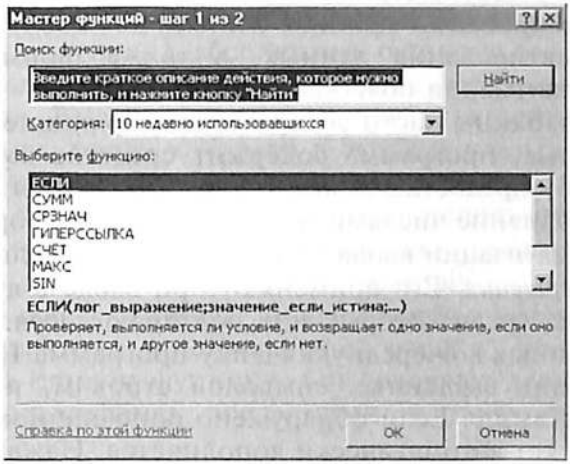


Рис. 5.6. Окно Мастера функций

После выбора функции открывается окно параметров, где нужно ввести необходимые аргументы.

Формулой в электронных таблицах называют арифметические и логические выражния. Начинается формула со знака =. Формулы могут содержать константу числа и текст, заключенный в кавычки, ссылки на ячейки, знаки арифметических, логических и других операций, встроенные функции, скобки, закладки и др. Для отображения формул, а не значений нужно использовать команды **Сервис → Параметры → Вид → Показывать → Формула**.

Функции могут вводиться в таблицу в составе формул либо отдельно.

Имена функций можно набирать в любом регистре, а формулы — только в верхнем. Тип входных данных, содержащихся в

каждой ячейке, определен первым символом, который трактуется как команда переключения режима.

Режимы работы Excel: готовность ввода данных; редактирование (**F2**); командный (**F10**). Для разделения строк в ячейке используется сочетание клавиш **Alt+Enter**.

Особенность электронных таблиц заключается в возможности применения формул для описания связи между значениями различных ячеек. Расчет по заданным формулам выполняется автоматически. Изменение содержимого какой-либо ячейки приводит к пересчету значений всех ячеек, которые с ней связаны формульными отношениями, и тем самым к обновлению всей таблицы в соответствии с изменившимися данными.

При формировании таблицы выполняют ввод, редактирование и форматирование данных, а также формул. Наличие средств автоматизации облегчает эти операции.

Так как таблицы часто содержат повторяющиеся или однотипные данные, программа содержит средства *автоматизации ввода*. К числу предоставляемых средств относятся автозавершение, автозаполнение числами и автозаполнение формулами.

Для автоматизации ввода текстовых данных используется *метод автозавершения*. Его применяют при вводе в ячейки одного столбца строк, среди которых есть повторяющиеся. В ходе ввода текстовых данных в очередную ячейку программа Excel проверяет соответствие введенных символов строкам, имеющимся в этом столбце выше. Если обнаружено однозначное совпадение, введенный текст автоматически дополняется. Нажатие на клавишу **Enter** подтверждает операцию автозавершения, в противном случае ввод можно продолжать, не обращая внимания на предлагаемый вариант.

Автозаполнение числами — процедура автоматического заполнения строки или столбца, изменение последних данных или постоянного значения с перетаскиванием.

В правом нижнем углу рамки текущей ячейки имеется черный квадратик — маркер заполнения. При наведении на него указатель мыши приобретает форму тонкого черного крестика. Перетаскивание маркера заполнения рассматривается как операция «размножения» содержимого ячейки в горизонтальном или вертикальном направлении. Если ячейка содержит число (в том числе дату, денежную сумму), то при перетаскивании происходит копирование ячеек или их заполнение арифметической прогрессией. Для выбора способа **Автозаполнение** следует произво-

дить специальное перетаскивание с использованием правой кнопки мыши.

Например, ячейка A1 содержит число 1. Наведите указатель мыши на маркер заполнения, нажмите правую кнопку мыши и перетащите маркер заполнения так, чтобы рамка охватила ячейки A1, B1 и C1, а затем отпустите кнопку мыши. Если теперь выбрать в открывшемся меню пункт **Копировать ячейки**, все ячейки будут содержать число 1. Если же выбрать пункт **Заполнить**, то в ячейках окажутся числа 1, 2 и 3.

Автозаполнение формулами выполняется так же, как и автозаполнение числами. Его особенность заключается в необходимости копирования ссылок на другие ячейки. Относительные ссылки изменяются в соответствии с относительным расположением копии и оригинала, абсолютные остаются без изменения.

Стандартные функции используются в программе Excel только в формулах. Вызов функции состоит в указании в формуле имени функции, после которого в скобках указывается список параметров. Отдельные параметры разделяются в списке точкой с запятой. В качестве параметра может использоваться число, адрес ячейки или произвольное выражение, для вычисления которого также могут использоваться функции.

В режиме ввода формулы в левой части строки формул, где раньше располагался номер текущей ячейки, появляется раскрывающийся список функций. Он содержит десять функций, которые использовались последними, а также пункт **Другие функции**.

При выборе пункта **Другие функции** запускается Мастер функций, облегчающий выбор нужной функции. В раскрывающемся списке **Категория** выбирается категория, к которой относится функция (если категорию определить затруднительно, используют пункт **Полный алфавитный перечень**), а в списке **Выберите функцию** — конкретная функция данной категории. После щелчка по кнопке **ОК** имя функции заносится в строку формул вместе со скобками, ограничивающими список параметров. Текстовый курсор устанавливается между этими скобками. Вызвать Мастер функций можно и проще — щелчком по кнопке **Вставка функции** в строке формул. Как только имя функции выбрано, на экране появляется диалоговое окно **Аргументы функции**. Это окно, в частности, содержит значение, которое получится, если немедленно закончить ввод формулы. Правила вычисления формул, содержащих функции, не отличаются от правил вычисления более про-

стых формул. Ссылки на ячейки, используемые в качестве параметров функции, также могут быть относительными или абсолютными, что учитывается при копировании формул методом **Автозаполнение**. Для вычислений в Excel применяют как основные средства программы, так и дополнительные (надстройки).

Итоговые вычисления предполагают получение числовых характеристик, описывающих определенный набор данных в целом. Например, возможно вычисление суммы значений, входящих в набор, среднего значения и других статистических характеристик, количества или доли элементов набора, удовлетворяющих определенным условиям. Проведение итоговых вычислений в программе выполняется с помощью встроенных функций. Особенность использования таких *итоговых функций* состоит в том, что при их задании программа пытается «угадать», в каких ячейках заключен обрабатываемый набор данных, и задать параметры функции автоматически. В качестве параметра итоговой функции обычно задается некоторый диапазон ячеек, размер которого определяется автоматически. Выбранный диапазон рассматривается как отдельный параметр (массив), и в вычислениях используются все ячейки, составляющие его.

Для итоговых вычислений наиболее типичной является функция *суммирования* (СУММ), для применения которой есть отдельная кнопка (**Автосумма**) на панели инструментов **Стандартная**. Диапазон автосуммирования включает ячейки с данными, расположенные над текущей ячейкой (предпочтительнее) или слева от нее и образующие непрерывный блок. При неоднозначности выбора используется диапазон, непосредственно примыкающий к текущей ячейке. Автоматический подбор диапазона не исключает возможности редактирования формулы. Можно переопределить диапазон, который был выбран автоматически, а также задать дополнительные параметры функции.

Функции для итоговых вычислений выбираются обычным образом, с помощью раскрывающегося списка в строке формул или с использованием Мастера функций. Все эти функции относятся к категории статистических. В их число входят функции ДИСП (вычисляет дисперсию), МАКС (максимальное число в диапазоне), СРЗНАЧ (среднее арифметическое значение чисел диапазона), СЧЕТ (подсчет ячеек с числами в диапазоне) и др.

Логические операции Excel позволяют при решении некоторых задач значение ячейки вычислять одним из нескольких способов

в зависимости от выполнения одного или нескольких условий (см. раздел 5.2.2).

Использование надстроек. *Надстройки* — это специальные средства, расширяющие возможности программы Excel. Хотя эти средства считаются внешними, дополнительными, доступ к ним осуществляется с помощью обычных команд строки меню (обычно через меню **Сервис** или **Данные**). Команда открывает специальное диалоговое окно, оформление которого не отличается от стандартных диалоговых окон программы Excel. Подключить или отключить установленные надстройки можно с помощью команды **Сервис → Надстройки**.

Надстройки, поставляемые вместе с программой Excel:

1. *Пакет анализа* (*Analysis ToolPak*). Обеспечивает дополнительные возможности анализа наборов данных. Выбор определенного метода анализа осуществляется в диалоговом окне **Data Analysis (Анализ данных)**, которое открывается командами **Сервис → Data Analysis (Анализ данных)**.

2. *Мастер суммирования* (*Conditional Sum Wizard*). Позволяет автоматизировать создание формул для суммирования данных в столбце таблицы. При этом ячейки могут включаться в сумму только при выполнении определенных условий. Запуск Мастера осуществляется с помощью команд **Сервис → Conditional Sum (Частичная сумма)**.

3. *Мастер подстановок* (*Lookup Wizard*). Автоматизирует создание формулы для поиска данных в таблице по названию столбца и строки. Мастер позволяет произвести однократный поиск или предоставляет возможность ручного задания параметров поиска. Вызывается командами **Сервис → Lookup (Поиск)**.

4. *Подбор параметра* — надстройка, используемая для решения задач оптимизации. Пример 5.3 см. ниже.

5. *Поиск решения* (*Solver Add-in*). Эта надстройка используется для решения задач оптимизации. Ячейки, для которых подбираются оптимальные значения и задаются ограничения, выбираются в диалоговом окне **Solver Parameters (Поиск решения)**, которое открывается при помощи команд **Сервис → Solver (Поиск решения)**.

6. *Построение диаграмм и графиков*. Выбор типа диаграммы, выбор данных, оформление диаграммы, размещение диаграммы, редактирование диаграммы.

Пример 5.1. Решение уравнений с помощью надстройки Подбор параметра.

Задача. Найдите решение уравнения $x^3 - 3x^2 + x = -1$.

Решение

1. Запустите Excel и откройте рабочую книгу, созданную ранее.

2. Создайте новый рабочий лист (**Вставка → Лист**), дважды щелкните по его ярлыку и присвойте ему имя Уравнение.

3. Занесите в ячейку A1 значение 0.

4. Занесите в ячейку B1 левую часть уравнения, используя в качестве независимой переменной ссылку на ячейку A1. Соответствующая формула может, например, иметь вид:

=A1^3-3*A1^2+A1.

5. Выполните команды **Сервис → Подбор параметра**.

6. В поле **Установить** в ячейке укажите B1, в поле **Значение** задайте –1, в поле **Изменяя значение ячейки** укажите A1.

7. Щелкните по кнопке **ОК** и посмотрите результат подбора, отображающий в диалоговом окне **Результат подбора параметра**. Щелкните по кнопке **ОК**, чтобы сохранить полученные значения ячеек.

8. Повторите расчет, задавая в ячейке A1 другие начальные значения, например 0,5 или 2. Совпали ли результаты вычислений? Чем можно объяснить различия? (При наличии нескольких корней результат решения уравнения зависит от того, какое число было выбрано в качестве начального приближения.)

9. Сохраните рабочую книгу.

Контрольные вопросы

1. Каковы назначение и основные функции процессора Excel?
2. Что такое рабочая книга и рабочие листы Excel?
3. Какие виды адресов ячеек имеет электронная таблица?
4. Что такое правило относительной адресации?
5. Какие вы знаете статистические и математические функции Excel?
6. Что называется формулой в Excel и из чего она состоит?
7. Какие в Excel имеются средства автоматизации?
8. Каково назначение Мастера функций?
9. Какие функции используются при решении логических задач?
10. Какие надстройки применяют для решения задач?

5.2.1. Заполнение таблиц в Excel, редактирование, форматирование, фильтрация элементов и данных

Ячейке можно присвоить *произвольное имя*, используя команды меню **Вставка** → **Имя** → **Присвоить**, а с помощью команды **Создать** — сформировать имя ячейки из заголовков соседних строк и столбцов. Выделив диапазон ячеек, присваивают им имена. В диалоговом окне флажками отмечают расположение заголовков, из которых создают имена. Вводится информация только в активную ячейку (в нее помещается курсор с помощью щелчка левой кнопки мыши). Вводимая (с клавиатуры или копированием) информация отражается не только в ячейке, но и в *строке формул*, расположенной над строкой заголовков столбцов. Если в ячейке отражается числовой результат, то в строке формул — формулы расчета в активной ячейке. Если в строке формул ввести данные, то они отобразятся в активной ячейке. В левой части строки формул при вводе возникают три кнопки: с красным крестиком — для прекращения процесса ввода и удаления текущих данных; с зеленой галочкой — для фиксации данных; со значком f_k или = — для вывода на экран диалогового окна **Мастер функций**.

Ввести ряд значений в диапазон ячеек можно, выполнив команду **Заполнить** из меню **Правка** или с помощью автозаполнения.

Ввод любой формулы начинается со знака =, означающего начало математической операции (например, формула для умножения двух чисел выглядит так: =15*20). Формулы могут содержать ссылки на ячейки (например, если в ячейке D5 записана формула =A2+C5-F1, то в ней будет помещаться результат вычисления суммы данных, находящихся в ячейках A2 и C5, и вычитания значения, находящегося в ячейке F1).

Примечания можно вставить, выполнив команду **Примечание** из меню **Вставка**, и далее заполнить все поля открывшегося диалогового окна.

Для внесения исправлений (редактирования) прежде всего необходимо выделить ячейку или диапазон ячеек. Выделение несмежных ячеек необходимо выполнять, удерживая нажатой клавишу **Ctrl**. Очистить или удалить ячейки можно, сначала выделив их, а затем нажав клавишу **Delete**. Форматы ячеек при этом сохраняются.

Чтобы переместить данные, необходимо их выделить, а затем последовательно использовать команды **Вырезать** → **Вставить**.

Для многократных вставок используют команду **Копировать**. Содержимое группы ячеек можно переместить перетаскиванием мышью, подведя курсор к внешней границе выделенной области. Курсор при этом принимает вид стрелки.

Чтобы добавить строку или столбец, следует использовать команды **Вставка → Строка** или **Вставка → Столбец**.

Основные *типы информации*, используемые в Excel:

- *число* (например, 3; 21; 507; 10 %);
- *текст* (например, Итого_2-й месяц);
- *дата и время* (например, Нояб-99; 14/10/01; 2:00 PM). Для ввода текущей даты в активную ячейку достаточно нажать комбинацию клавиш **Ctrl+;**;
- *формула* (например, = СУММ(F3:F8) или C5*3,22).

Можно также вводить *примечания* (например, «Excel хранит и выводит на экран элементы каждого типа данных по-разному»).

Числа можно записывать в форме:

- обычной;
- экспоненциальной (с плавающей точкой), если число слишком большое или маленькое (имеет большую разрядность). Форма записи: x= M*Eрх, где M — мантисса, цифровая часть, E — основание системы счисления (для десятичных чисел E=10), рх, — порядок (степень), на который надо умножить M, чтобы получить число. Например:
  1E-3=1*10-3=0,001;
  0.257E+3=257.

Текст состоит из букв, цифр, пробелов. Например: «10 Мб».

Форматирование таблиц. При форматировании в Excel можно изменять ширину столбцов и высоту строк, присваивать ячейкам различные форматы данных, вертикально выравнивать данные в ячейках и т. д. В стандартной ячейке можно разместить только 8 цифр при использовании шрифта Arial размером 10. Если этого недостаточно, то ширину и высоту ячеек изменяют с помощью соответствующих команд **Автоподбор ширины**, **Строки**, **Столбцы** меню **Формат**.

Для форматирования ячеек следует выбрать команду **Ячейки** из меню **Формат**, а далее использовать вкладки открывшегося диалогового окна **Формат ячеек**. Следует использовать один из способов форматирования ячеек таблицы. Например, перетаскивание правой границы заголовка столбца до нужной ширины и нижней границы заголовка строки до нужной высоты или с по-

мощью команд меню **Формат** → **Столбец** или **Формат** → **Строка**, предварительно выделив необходимые листы. Затем в открывшемся подменю выбрать пункт **Ширина** и ввести необходимое значение, аналогично для строк — в открывшемся подменю выбрать пункт **Высота** и ввести необходимое значение.

В современные версии Excel включено более 400 функций из разных областей деятельности. Каждая функция вызывается в соответствии с определенным способом записи обращения к ней. Этот способ обычно представлен определенным диалоговым окном. В нем можно воспользоваться подсказкой относительно каждой функции (с примерами). Специальная программа Мастер функций, которую можно вызвать командой **Функция** из меню **Вставка** или активизацией кнопки со значком f_x на панели инструментов, позволяет просмотреть девять категорий функций и выбрать нужный вид вычислений. В левом разделе окна Мастера функций следует выбрать категорию функции из списка **Категория**, а в правом — нужную функцию. Затем после двойного щелчка по имени выбранной функции (или по кнопке **Далее**) откроется второе диалоговое окно для ввода аргументов функции. Вводят все необходимые значения аргументов. Щелкнув по кнопке **Готово** или **ОК**, получают в активной ячейке значение выбранной функции.

Дублирование (или *копирование*) формулы в соседние ячейки производят с помощью команды **Заполнить** меню **Правка** или перетаскиванием мышью.

Фильтрация — наиболее быстрый способ получить необходимые данные. Чтобы создать фильтр, следует выбрать команды меню **Данные** → **Фильтр** → **Автофильтр** или **Расширенный фильтр** и указать необходимые условия для отбора.

На примере решения задачи ниже показано выполнение работы по заполнению таблиц, редактированию, форматированию, созданию фильтров и простых вычислений в таблицах.

Пример 5.2. Некая фирма, торгующая мороженым в городе *H*, ведет учет выручки в миллионах рублей по 4 округам города в летние месяцы 2000 г.

Исходные данные: 12 чисел, каждое из которых — выручка по конкретному округу, за конкретный месяц.

На основе данных найти:

- сумму выручки по городу за каждый месяц;
- сумму выручки по каждому округу за все лето;

- общую сумму выручки;
- процент выручки по каждому округу относительно общей суммы.

Последовательность действий:

1. Вводим исходные данные.
2. Выполняем требуемые операции и расчеты.
3. Выполняем фильтрацию данных.

Результаты представлены на рис. 5.7.

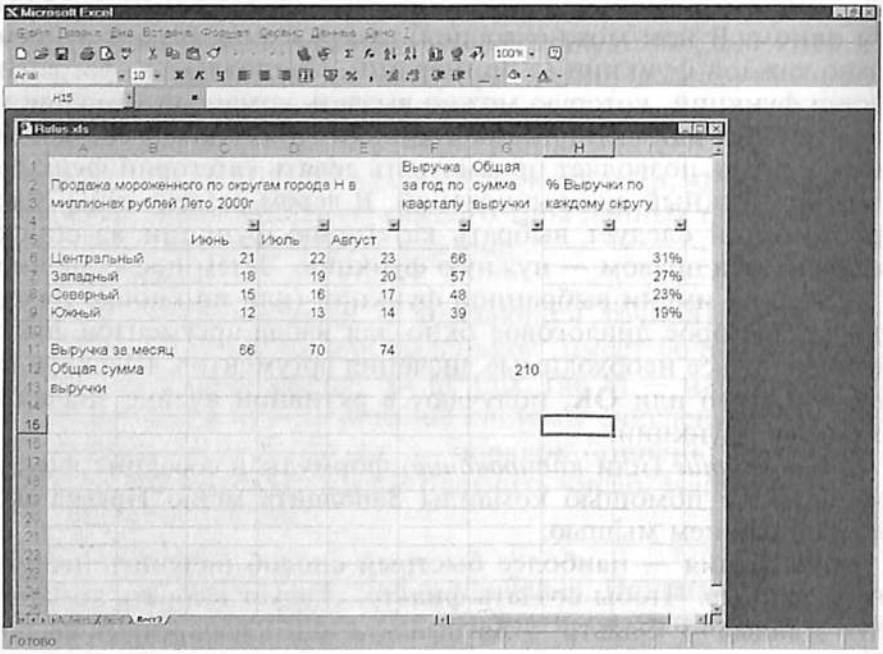


Рис. 5.7. Заполнение таблиц, редактирование, форматирование, создание фильтров и простые вычисления в таблицах

Контрольные вопросы

1. Каковы основные типы информации, используемые в электронных таблицах?
2. Каковы способы изменения ширины столбцов и высоты строк?
3. Какие способы задания заголовков в Excel вы знаете?
4. Как в Excel отличить текст от числа и формулы?
5. Как увидеть формулу, записанную в ячейку, и сделать так, чтобы в ячейке отображался не результат вычислений, а сама формула?

5.2.2. Создание списков. Расчеты с использованием логических операций в Excel

Для обработки больших объемов информации (например, списков телефонов, адресов, организаций и т. п.) в Excel имеются средства их обработки: таблицы, разделенные на *столбцы-поля* и *строки-записи*. *Список* — набор строк с постоянными заголовками столбцов и последовательными данными, по существу, представляет собой *базу данных* — организованное хранилище информации. Список должен иметь постоянное количество полей (столбцов). Имя столбца называется меткой. Столбец должен содержать однотипную информацию и не содержать пустые ячейки. Количество строк должно быть переменным, чтобы можно было добавлять, удалять, переставлять записи.

При создании списка необходимо выполнять следующие операции:

- открыть новый лист;
- создать заголовки;
- отформатировать ячейки под заголовками;
- ввести записи под заголовками;
- сохранить книгу на диске.

Автозаполнение выполняют при вводе повторяющихся значений, выбрав команды **Сервис** → **Параметры**. В диалоговом окне **Параметры** выбирают вкладку **Правка** и ставят флажок **Автозаполнение значений ячеек**.

При создании списков удобно использовать специальные формы. Для этого следует выбрать команды меню **Данные** → **Форма**. В диалоговом окне набирают новые записи и добавляют их в список. Сортировку данных выполняют с помощью команды **Сортировка** и вкладок диалогового окна, фильтрацию — с помощью команд **Фильтр** и **Автофильтр**.

Для новой организации полей на листе выбирают команду **Сводная таблица** из меню **Данные**. Далее нужно следовать указаниям в открывшемся окне **Мастер сводных таблиц**.

Логические функции применяются в тех случаях, когда при решении задачи могут возникнуть некоторые условия, например, товар задан в килограммах, а цена — в рублях за тонну или за штуку.

Для расчетов в таблице с использованием логических операций применяют условную функцию ЕСЛИ. Она имеет следующий формат: ЕСЛИ<логическое выражение>, <выражение1>,

<выражение2>. Первый аргумент функции принимает одно из двух значений: «истина» или «ложь». Функция ЕСЛИ возвращает обычно число или строку символов, а И или ИЛИ возвращает логическое значение «истина» или «ложь».

Условная функция ЕСЛИ возвращает:

ЕСЛИ (лог._выражение; значение1 _если_истина; значение2_если_ложь).

Функция ИЛИ имеет следующий вид:

ИЛИ (значение1; значение2; ...).

Функция И имеет следующий вид:

И (значение1; значение2; ...).

Если условий много, записывать вложенные функции ЕСЛИ становится неудобно. В этом случае вместо ЕСЛИ можно указать одну из двух логических функций: ИЛИ (OR), И(AND). Функция И принимает значение «истина», если одновременно истинны все выражения.

Например, продавец торговой точки ежедневно составляет отчет о продажах. Для этого он использует таблицу, подготовленную в Excel (рис. 5.8).

Название товара			Продано, кг	Цена, руб./кг	Стоимость, руб.	
Сахар-песок			50	25		
Крупа манная			20	15		
Соль "Экстра"			10	10		
Рис		1	20	30		
Всего			100			

Рис. 5.8. Отчет о продажах за 10 июня 2009 г. Таблица Excel

Как видно из рис. 5.8, товар отпускается килограммами и цена его задана в руб./кг. На самом деле торговая точка может продавать товар и килограммами, и тоннами, и штуками, а цена может задаваться и в руб./кг, и в руб./шт, и в других единицах.

Пример 5.3. Предположим сначала, что количество товара в таблице задается либо в килограммах, либо в тоннах, а цена — в рублях за 1 кг. Для правильного расчета стоимости в этом случае необходимо установить, в каких единицах задано количество

продукта, и в зависимости от результата использовать ту или иную формулу.

Решение. Если количество задано в килограммах, стоимость $C = Q \cdot Ц$, где Q — количество килограммов; Ц — цена (руб./кг). Если количество задано в тоннах, стоимость рассчитывается по формуле $C = Q1 \cdot Ц$, где $Q1$ — количество товара (т) = $Q \cdot 1000$.

Пусть в ячейке C5 помещается код единицы измерения количества продукции, который принимает следующие значения: 1 — (кг); 2 — (т).

В ячейке D5 помещается количество продукта, в ячейке E5 — цена (руб./кг). В ячейку F5 необходимо поместить стоимость товара. Тогда в эту ячейку можно записать функцию =ЕСЛИ (C5=1, D5*E5, D5*1000*E5) — рис. 5.9.

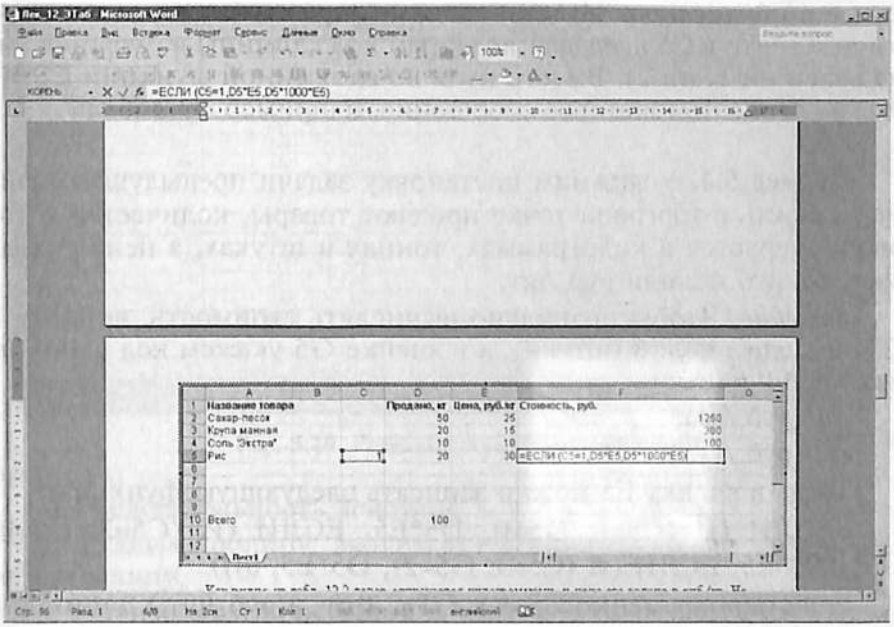


Рис. 5.9. Расчет стоимости товара по формуле

Здесь логическое выражение C5=1 (в данном случае это просто условное выражение). Если в C5 записана 1, условие выполнено и значение этого выражения равно «истине». Поэтому функция ЕСЛИ, записанная в ячейке F5, принимает значение D5*E5 (т. е. $C = Q \cdot Ц$). Если значение логического выражения — «ложь» (C5 не равно 1), функция ЕСЛИ возвратит значение D5*1000*E5.

Здесь намеренно допущена небрежность, *подмена* условия задачи: цена будет умножаться на тысячу не при C5 = 2, а при C5 не равно 1. Если пользователь ошибается и запишет в C5 не 2, а, скажем, 3 или 11, Excel все равно будет считать, что товар отпускается в тоннах. Поэтому на место третьего аргумента надо записать вложенную функцию ЕСЛИ:

=ЕСЛИ (C5=l, D5*E5, ЕСЛИ (C5=2, D5*1000*E5,0)).

В этом случае, если C5 не равно 1, функция вернет значение вложенной функции ЕСЛИ, которое, в свою очередь, зависит от выполнения условия C5 = 2.

Если это условие выполнено (т. е. единица измерения — тонна), значением функции будет D5*1000*E5, если нет — значением функции будет нуль. Иными словами, нулевая стоимость товара означает, что в C5 *неправильно* указан код единицы измерения (он не равен ни 1, ни 2). Вместо нуля во вложенной функции ЕСЛИ можно записать текстовое сообщение, например «Ошибка!».

Пример 5.4. Усложним постановку задачи предыдущего примера. Пусть в торговой точке продают товары, количество которых измеряется в килограммах, тоннах и штуках, а цена указывается в руб./кг или руб./шт.

Решение. Чтобы правильно вычислять стоимость, добавим в C5 еще один код: 3 (штуки), а в ячейке G5 укажем код единицы измерения цены:

- 1 (руб./кг);
- 2 (руб./шт.).

Тогда в ячейку F5 можно записать следующую функцию:

=ЕСЛИ (И (C5=1, G5=l), D5*E5, ЕСЛИ (И (C5=2, G5=l), D5*1000*E5, ЕСЛИ (И (C5=3, G5=2), D5*E5, 0))).

Если одновременно C5=1 и G5=2 (шт. и руб./шт.), стоимость равна D5*E5; если одновременно C5=2 и G5=l (т и руб./кг), стоимость равна D5*1000*E5. Нулевая стоимость товара означает, что либо неправильно указан хотя бы один из кодов (например, G5=4), либо введено недопустимое сочетание кодов (например, товар в кг, а цена в руб./шт.).

Подобным же образом можно использовать и функцию ИЛИ.

В Excel предусмотрены также еще три логические функции (НЕ, ЛОЖЬ и ИСТИНА), которые здесь не рассматриваются.

Пример 5.5. Магазин продал продукты (список см. в таблице на рис. 5.10). Нужно рассчитать:

1. Выручку с продажи за килограмм в рублях.
2. Выручку с продажи за килограмм в евро.
3. Выручку с продажи за тонну в рублях.
4. Выручку с продажи за тонну в евро.

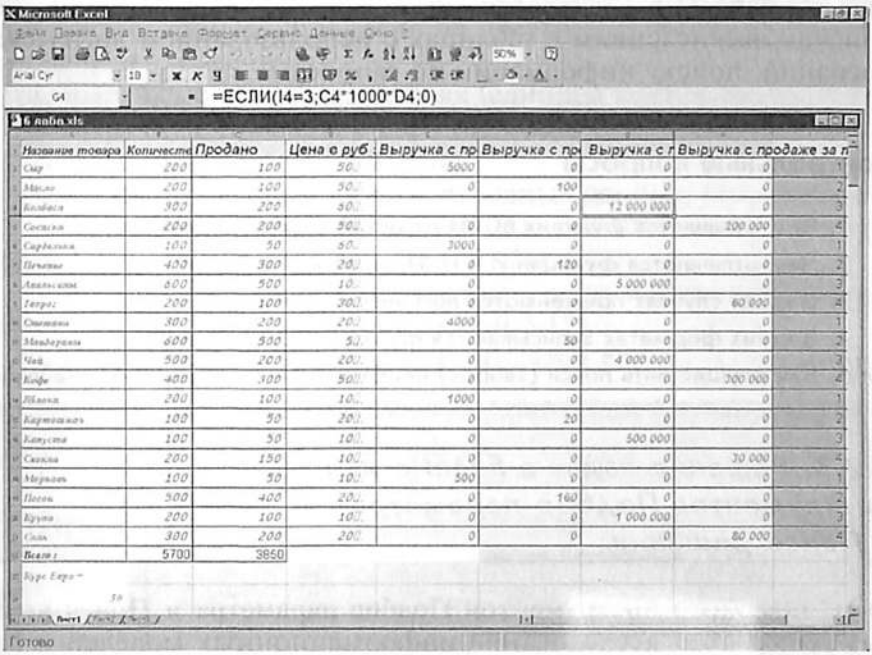


Рис. 5.10. Результаты решения задачи

Последовательность действий:

1. Создаем таблицу, вводим список исходных данных, сохраняем таблицу.

2. Производим простейшие расчеты в таблице с использованием логических операций. Для правильного расчета устанавливаем, в каких единицах задано количество товара, а также в каких денежных единицах рассчитывается стоимость товара.

При решении данной задачи возникают некоторые условия. Товар может рассчитываться в разных единицах измерения.

Чтобы узнать определенную выручку, в последней колонке нужно ввести коды.

1 — выручка с продаж за килограмм в рублях.

2 — выручка с продаж за килограмм в евро.

3 — выручка с продаж за тонну в рублях.

4 — выручка с продаж за тонну в евро.

Запросы необходимых данных можно выполнить, применяя команды **Автофильтр** или **Расширенный фильтр**.

Решив предлагаемые задачи, вы приобретете практические навыки работы с электронными таблицами Excel по созданию списков, вычислениям в таблицах с использованием логических операций, поиску информации по запросам.

Контрольные вопросы

1. Чем отличается функция ЕСЛИ от остальных функций?
2. Чем отличаются функции И и ИЛИ от функции ЕСЛИ?
3. В каких случаях применяются логические функции?
4. В каких форматах записываются функции ЕСЛИ, И, ИЛИ?
5. Как осуществить поиск (запрос) необходимой информации в таблице?

5.2.3. Решение задач в Excel с использованием инструментов Подбор параметра и Поиск решения

С помощью инструментов **Подбор параметра** и **Поиск решения** выполняют исследование информационных моделей и вычисления.

При вычислениях с помощью инструмента **Подбор параметра** (вызывается командами **Сервис → Подбор параметра**) требуемое значение получают в *целевой ячейке* путем изменения значения влияющей ячейки. Целевая ячейка должна ссылаться на ячейку с изменяемым значением.

При вычислениях с помощью инструмента **Поиск решения** (вызывается командами **Сервис → Надстройки → Поиск решения**) результат получают на основе изменения значений нескольких ячеек. Кроме того, можно задать условия — ввести ограничения (например, -цел — целое). При решении в диалоговом окне **Поиск решения** следует установить параметры в целевой ячейке, а в поле **Изменяя ячейки** ввести абсолютные адреса изменяемых ячеек. Ввести ограничения. В диалоговом окне **Результаты поиска**

решения устанавливают тип отчета **Результаты**. Сохраняют найденное решение.

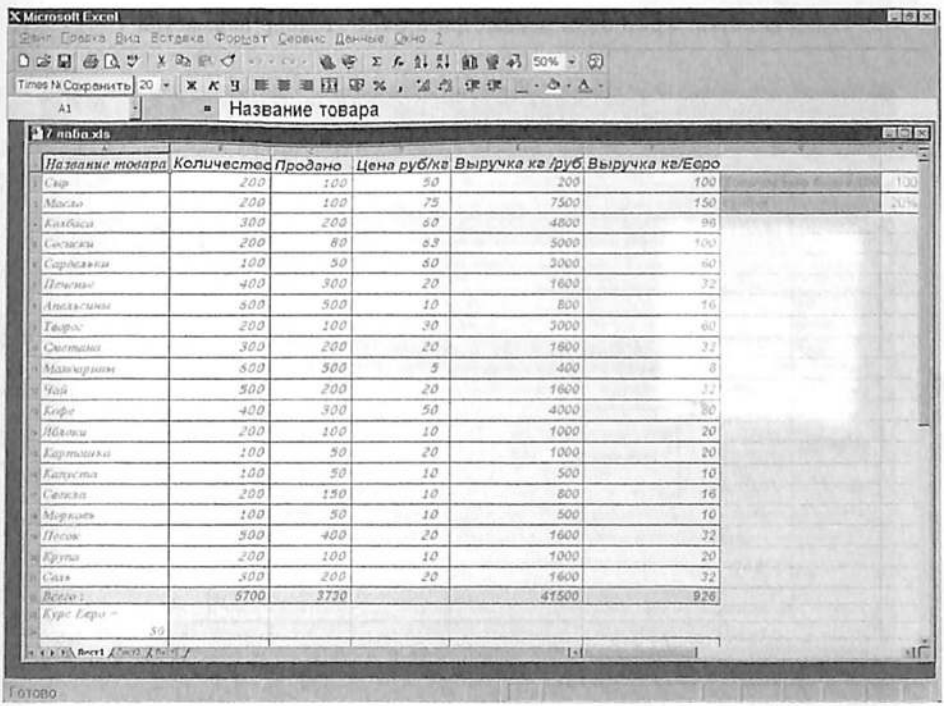
Для расчетов в таблице с использованием инструментов **Подбор параметра** и **Поиск решений в электронных таблицах Excel**, рассмотрим решение следующей задачи.

Магазин продал продукты (перечень см. в таблице на рис. 5.11). Если количество товара превышало 100 штук, то предоставлялась скидка 20 %, если не превышало, то скидка не предоставлялась. Рассчитать:

1. Выручку с продажи в рублях (единица измерения: кг/руб.).
2. Выручку с продажи в евро (единица измерения: кг/евро).

Последовательность действий:

1. В таблицу вводим список исходных данных, сохраняем ее.
2. Производим расчеты в таблице с использованием инструмента **Подбор параметра** (см. рис. 5.12 и 5.13).
3. Выполняем расчеты в таблице с использованием инструмента **Поиск решения** (см. рис. 5.14 и 5.15).
4. Сохраняем документ. Распечатываем его.



Название товара	Количество	Продано	Цена руб/кг	Выручка кг/руб	Выручка кг/Евро
Сыр	200	100	50	200	100
Масло	200	100	75	7500	150
Колбаса	300	200	60	4800	96
Сосиски	200	80	63	5000	100
Сардельки	100	50	60	3000	60
Печенье	400	300	20	1600	32
Апельсины	800	500	10	800	16
Творог	200	100	30	3000	60
Сметана	300	200	20	1600	32
Мандарины	800	500	5	400	8
Чай	500	200	20	1600	32
Кефир	400	300	50	4000	80
Яблоки	200	100	10	1000	20
Картошка	100	50	20	1000	20
Капуста	100	50	10	500	10
Свекла	200	150	10	800	16
Морковь	100	50	10	500	10
Песок	800	400	20	1600	32
Крупа	200	100	10	1000	20
Сахар	300	200	20	1600	32
Всего:	5700	3730		41500	926
Курс Евро =					
50					

Готово

Рис. 5.11. Исходные данные

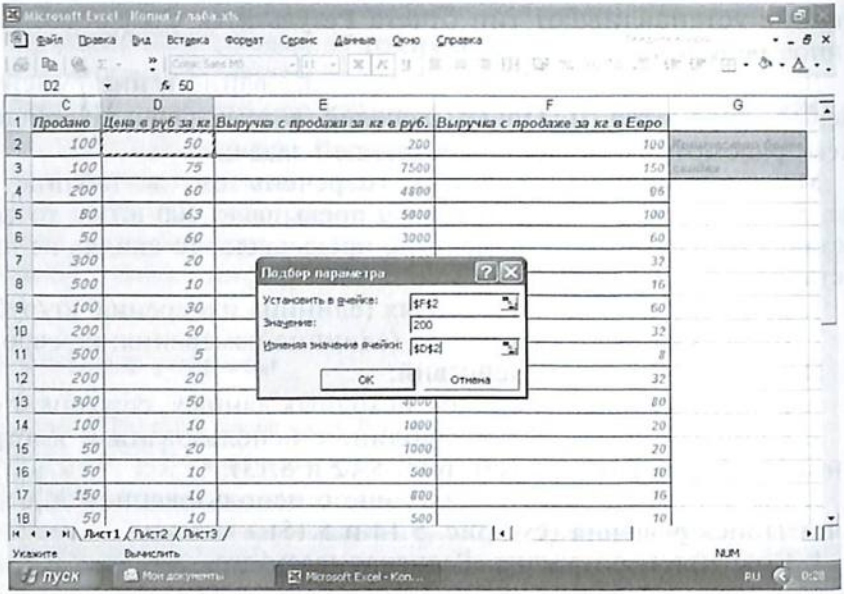


Рис. 5.12. Установка параметра в окне **Подбор параметра**

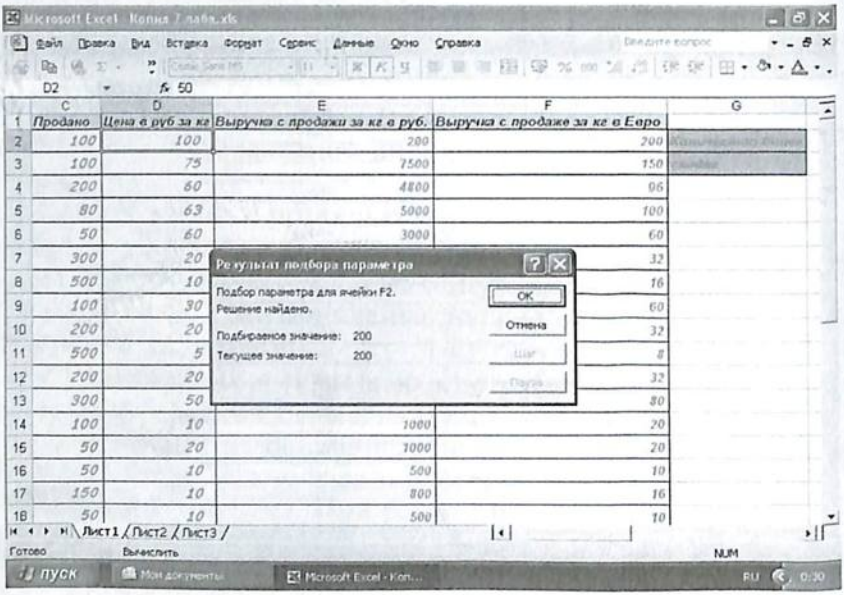


Рис. 5.13. Решение задачи с помощью инструмента **Подбор параметра**

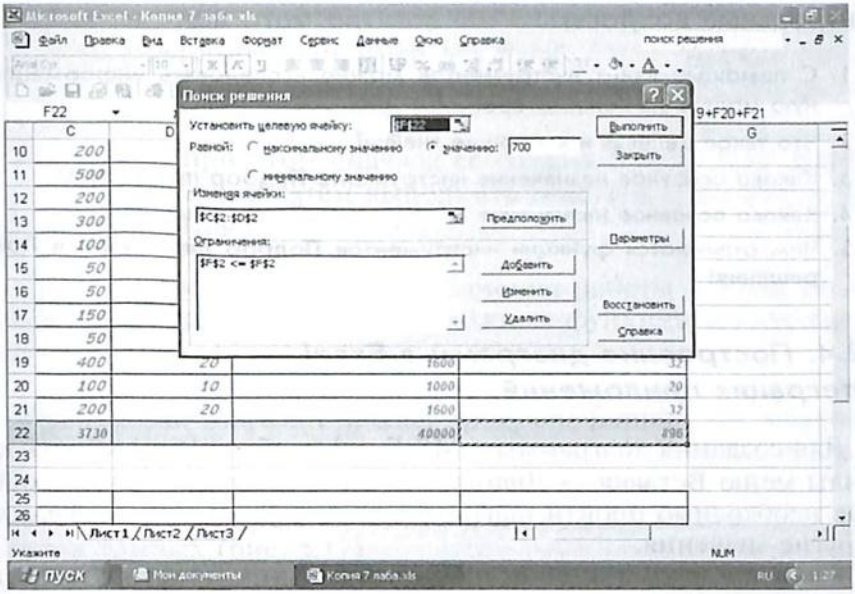


Рис. 5.14. Установка параметров в диалоговом окне **Поиск решения**

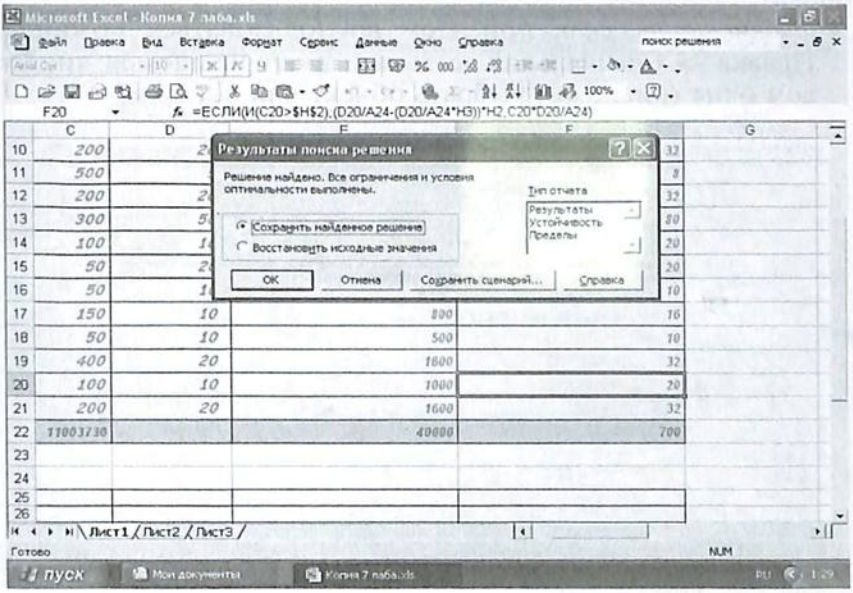


Рис. 5.15. Решение задачи с помощью инструмента **Поиск решения**

Контрольные вопросы

1. С помощью каких инструментов можно исследовать информационную модель на компьютере?
2. Что такое целевая и влияющие ячейки?
3. Каково основное назначение инструмента **Подбор параметра**?
4. Каково основное назначение инструмента **Поиск решения**?
5. Чем отличаются функции инструментов **Подбор параметра** и **Поиск решения**?

5.2.4. Построение диаграмм в Excel. Интеграция приложений

Для создания диаграммы в MS Excel необходимо выбрать команды меню **Вставка → Диаграмма**. В открывшемся диалоговом окне необходимо пройти шаги от 1 до 4, выбрав тип диаграммы и другие значения.

Сложная функция состоит из нескольких простых или вычисляется.

При интеграции приложений следует в документ MS Excel вставить:

- *фрагмент текста*, при этом воспользоваться командами **Правка → Специальная вставка** и в открывшемся диалоговом окне (рис. 5.16) выбрать объект для вставки;

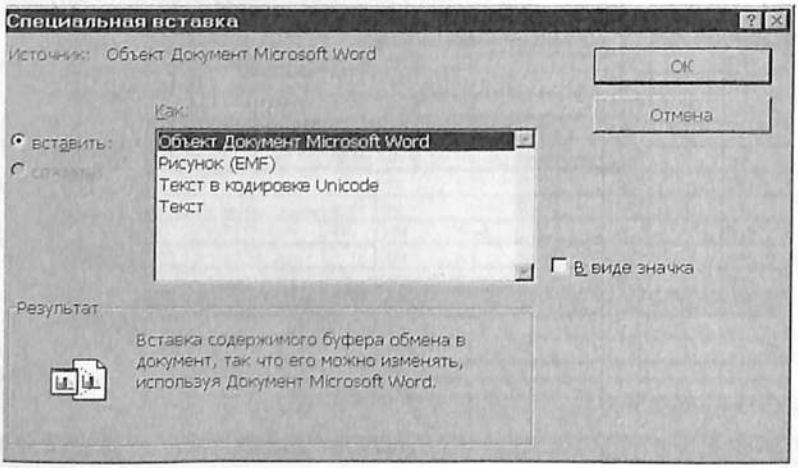


Рис. 5.16. Диалоговое окно **Специальная вставка**

- *рисунок*, при этом сначала его создать в любом графическом редакторе, например MS Paint. Далее рисунок скопировать в буфер обмена. В MS Excel выбрать объект для вставки;
- *формулу*, при этом сначала ее создать в редакторе формул MS Equation, а затем выполнить действия аналогично указанным выше.

Далее приведен пример выполнения работы в Excel по построению диаграмм, созданию сложных функций и интеграции приложений.

Пример 5.6. Магазин продал продукты (список см. в таблице на рис. 5.17). Нужно построить диаграммы наличия и продажи товара и рассчитать выручку с продажи за сыр.

1. Создаем диаграммы наличия и продажи товара на основе таблицы данных (рис. 5.17).

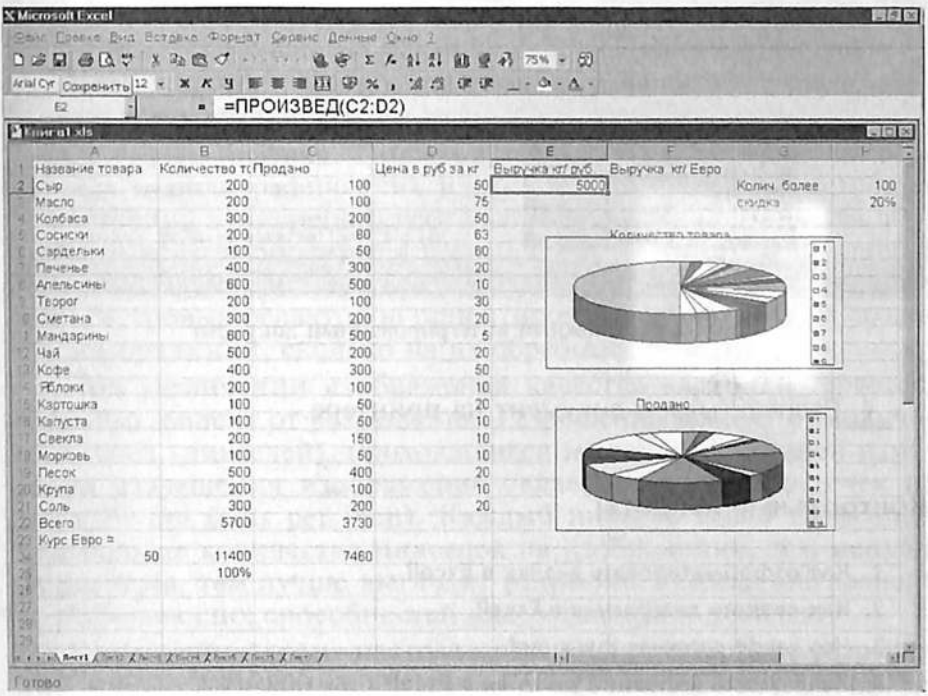


Рис. 5.17. Создание диаграмм

2. Создаем сложную функцию для определения выручки за продажу сыра (см. рис. 5.17).

3. Выполняем интеграцию приложений для создания сложного документа (рис. 5.18) с помощью диалогового окна **Специальная вставка**.

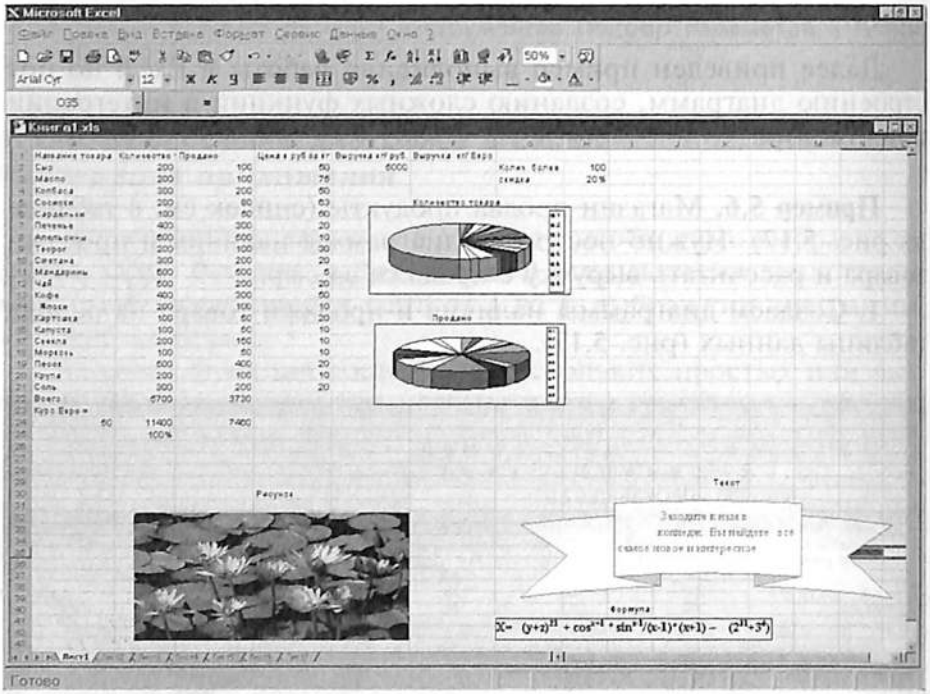


Рис. 5.18. Сложный интегрированный документ

4. Распечатываем документ на принтере.

Контрольные вопросы

1. Как отформатировать ячейку в Excel?
2. Как создать диаграмму в Excel?
3. Что такое сложная функция?
4. Как создать (вставить) рисунок в Excel?
5. Для чего служит функция **Специальная вставка**?

5.3. Технологии обработки графической информации. Технология мультимедиа

Работа с компьютерной графикой — одно из самых популярных направлений при использовании компьютеров. Не только профессиональные художники, но и обычные пользователи увлекаются этим занятием. Без компьютерной графики не обходится ни одна современная мультимедийная программа, и работа над графикой занимает часто до 90 % рабочего времени программистских коллективов по выпуску программ массового применения.

Необходимость широкого использования программных графических средств возникла вследствие развития интернет-технологий и, в частности, благодаря службе WWW. При оформлении сайтов каждый из них снабжается компьютерной графикой. Потребность в разработке привлекательных веб-сайтов во много раз превышает возможности профессиональных художников и дизайнеров. Поэтому графические средства разрабатывают с таким расчетом, чтобы предоставлять возможность для работы тем, кто не имеет профессиональных навыков и способностей к рисованию.

Различают три вида компьютерной графики: растровую, векторную и фрактальную.

Растровая графика. Применяется при разработке мультимедийных и полиграфических изданий. Для создания растровых иллюстраций чаще используют сканер, но в наше время для ввода растрового изображения в компьютер используют цифровые фотоаппараты и видеокамеры. Поэтому большинство графических растровых редакторов ориентированы не столько на создание изображений, сколько на их обработку.

При увеличении изображения качество картинки теряется. Качество зависит от разрешения. Разрешение зависит от количества точек (пикселей), приходящихся на длину. Единицей измерения разрешения изображения является количество точек на дюйм — dpi (dots per inch). Каждый пиксель имеет свой цвет. Чем больше количество пикселей на изображении, тем меньше их размеры, тем лучше выглядит результат. Разрешение зависит от разрешающих способностей видеоадаптера и дисплея.

Разрешение экрана настраивается ОС, стандартные разрешения — 640 × 480; 800 × 600; 1024 × 768; 1280 × 1024 точек и др. по горизонтали и вертикали соответственно.

Например, у экрана диагональю 15 дюймов разрешение изображения равно 28 × 21 см. 1 дюйм равен 25,4 мм, значит, разрешение — 800 × 600 пикселей.

При печати разрешение должно быть намного выше (72 dpi). Полиграфическая печать полноценного изображения требует разрешения 200—300 dpi. Например, стандартный фотоснимок 10 × 15 см должен содержать примерно 1000 × 1500 пикселей. А если изображение цветное (каждая точка использует 3 байта), то обычной цветной фотографией будет массив данных свыше 4 Мб. Большие объемы данных — это основная проблема. Увеличение размера точек растра визуально искажает иллюстрацию и делает ее грубой. Этот эффект называется *пикселизацией*.

Векторная графика. Предназначена для создания иллюстраций и в меньшей степени для их обработки. Оформление работ основывается на применении шрифтов и простейших геометрических элементов. При увеличении изображения качество картинки не изменяется.

В векторной графике основным элементом изображения является линия (неважно, какая: кривая или прямая). В векторной графике объем памяти, занимаемый линией, не зависит от ее размеров, поскольку линия представлена как формула (в виде нескольких параметров). Линия — это элементарный объект (примитив) векторной графики. Все, что есть в векторной графике, состоит из линий. Простые объекты объединяют в более сложные. Векторную графику часто называют *объектно-ориентированной графикой*. Объекты векторной графики хранятся в памяти в виде параметров, но на экран они выводятся в виде точек. Перед выводом на экран каждого объекта программа производит вычисление координат экранных точек изображения объекта, поэтому векторную графику называют *вычисляемой графикой*. Как и все объекты, линии имеют свойства: форму, характер линии, цвет и заполнение.

Иногда вместо понятия «линия» используют понятие «контур». Этот термин более полно отражает суть, поскольку контур может иметь любую форму — прямой, кривой, ломаной линии, фигуры.

Каждый контур имеет две или более *опорные точки*, также именуемые *узлами*. *Элемент контура*, заключенный между двумя смежными опорными точками, называют сегментом контура. *Форму* контура меняют перемещением опорных точек, добавле-

нием новых и удалением имеющихся узлов. Контур может быть *открытым* или *замкнутым* — когда последняя опорная точка одновременно является и первой. Свойства замкнутых и открытых контуров различны.

Контур служит элементарным графическим *объектом*. Из контуров создают новые объекты или их группы. С несколькими контурами выполняют операции *группировки*, *комбинирования*, *объединения*. В результате образуются соответственно *группа объектов*, *составной контур*, *новый контур*. После операции группировки каждый контур сохраняет свои свойства и принадлежащие ему узлы. После операции комбинирования составной контур приобретает новые свойства, но узлы остаются прежними. После операции объединения образуются новые и меняются свойства исходных контуров — узлы.

Фрактальная графика. Использует программные средства, предназначенные для автоматической генерации изображений путем математических расчетов. Создание фрактальной художественной композиции состоит не в рисовании и оформлении, а в программировании. Это вычисляемая графика, но отличается от векторной тем, что никакие объекты в памяти компьютера не хранятся.

Изображения строятся по уравнениям или системам уравнений, поэтому кроме формулы хранить ничего не надо. Изменив коэффициент в уравнении, можно получить совсем другую картинку. Простейший фрактальный объект — фрактальный треугольник. Фигуры последующих поколений сохраняют свойства предыдущих. Процесс наследования может продолжаться бесконечно. С помощью этой графики модулируют образцы живой природы, часто используемые для автоматической генерации иллюстраций.

В компьютерной графике применяют по меньшей мере три десятка *форматов файлов* для хранения изображений. Но лишь часть из них стала стандартом де-факто и применяется в подавляющем большинстве программ. Как правило, несовместимые форматы ассоциированы с файлами растровых, векторных, трехмерных изображений, хотя существуют форматы, позволяющие хранить данные разных классов. Многие приложения ориентированы на следующие специфические форматы.

Формат *TIFF* (*Tagged Image File Format*) — для хранения растровых изображений высокого качества (расширение имени файла — .tif). Относится к числу широко распространенных, от-

личается переносимостью между платформами (IBM PC и Apple Macintosh), обеспечен поддержкой со стороны большинства графических и дизайнерских программ, а также программ верстки. Предусматривает широкий диапазон цветового охвата — от монохромного черно-белого до 32-разрядной модели цветоделения CMYK. Начиная с версии 6.0, в формате TIFF хранят сведения о масках изображений. Для уменьшения размера файла применяется встроенный алгоритм сжатия LZW.

Формат *PSD* (*PhotoShop Document*) — собственный формат программы Adobe Photoshop (расширение имени файла — .psd), один из наиболее мощных по возможностям хранения. Позволяет запоминать параметры слоев, каналов, степени прозрачности, множества масок. Поддерживаются 48-разрядное кодирование цвета, цветоделение и различные цветовые модели. Основной недостаток выражен в том, что отсутствие эффективного алгоритма сжатия информации приводит к большому объему файлов.

Формат *PCX* появился как формат хранения растровых данных программы PaintBrush фирмы Z-Soft (расширение имени файла — .pcx). Отсутствие возможности хранить цветоделенные изображения, недостаточность цветовых моделей и другие ограничения привели к утрате популярности формата. К настоящему времени устарел.

Формат *PhotoCD* разработан фирмой Kodak для хранения цифровых растровых изображений высокого качества (расширение имени файла — .pcd). Сам формат хранения данных в файле называется Image Pac. Файл имеет внутреннюю структуру, обеспечивающую хранение изображения с фиксированными величинами разрешений, и потому размеры любых файлов лишь незначительно отличаются друг от друга и находятся в диапазоне 4—5 Мб. Каждому разрешению присвоен свой уровень, отсчитываемый от так называемого базового (Base), составляющего 512 × 768 точек. Всего в файле пять уровней — от Base/16 (128 × 192 точек) до Base-16 (2048 × 3072 точек). При первичном сжатии исходного изображения применяется метод субдискретизации практически без потери качества. Затем вычисляются разности Base — Basex4 и Basex4 — Basex16. Итоговый результат записывается в файл. Чтобы воспроизвести информацию с высоким разрешением, производится обратное преобразование. Для хранения информации о цвете использована цветовая модель YCC.

Формат *Windows Bitmap* служит для хранения растровых изображений в ОС Windows (расширение имени файла — .bmp).

Соответственно, поддерживается всеми приложениями, работающими в этой среде.

Формат *JPEG* (*Joint Photographic Experts Group*) предназначен для хранения растровых изображений (расширение имени файла — .jpg). Позволяет регулировать соотношение между степенью сжатия файла и качеством изображения. Применяемые методы сжатия основаны на удалении «избыточной» информации, поэтому формат рекомендуют использовать только для электронных публикаций.

Формат *GIF* (*Graphics Interchange Format*) стандартизирован в 1987 г. как средство хранения сжатых изображений с фиксированным (256) количеством цветов (расширение имени файла — .gif). Получил популярность в Интернете благодаря высокой степени сжатия. Последняя версия формата GIF89a позволяет выполнять чересстрочную загрузку изображений и создавать рисунки с прозрачным фоном. Ограниченные возможности по количеству цветов обусловливают его применение исключительно в электронных публикациях.

Формат *PNG* (*Portable Network Graphics*) — формат хранения изображений, предназначенный для их публикации в Интернете (расширение имени файла — .png). Создавался как замена для форматов GIF и JPEG. Поддерживаются три типа изображений — цветные с глубиной 8 или 24 бит и черно-белое с градацией 256 оттенков серого. Сжатие информации происходит практически без потерь, предусмотрены 254 уровня альфа-канала, чересстрочная развертка. Массового распространения не получил.

Формат *WMF* (*Windows MetaFile*) предназначен для хранения векторных изображений ОС Windows (расширение имени файла — .wmf). Поддерживается всеми приложениями этой системы. Однако отсутствие средств для работы со стандартизированными цветовыми палитрами, принятыми в полиграфии, и другие недостатки ограничивают его применение.

Формат *EPS* (*Encapsulated PostScript*) служит для описания как векторных, так и растровых изображений на языке PostScript фирмы Adobe, фактическом стандарте в области допечатных процессов и полиграфии (расширение имени файла — .eps). Так как PostScript является универсальным, в файле могут одновременно храниться векторная и растровая графика, шрифты, контуры обтравки (маски), параметры калибровки оборудования, цветовые профили. Для отображения на экране векторного содержимого

используется формат WMF, а растрового — TIFF. Но экранная копия лишь в общих чертах отображает реальное изображение (является недостатком EPS). Действительное изображение можно увидеть лишь на выходе выводного устройства, с помощью специальных программ просмотра или после преобразования файла в формат PDF в приложениях Acrobat Reader, Acrobat Exchange.

Формат *PDF* (*Portable Document Format*) позволяет описывать документы, разработанные фирмой Adobe (расширение имени файла — .pdf). Хотя этот формат в основном предназначен для хранения документа целиком, его впечатляющие возможности позволяют обеспечить эффективное представление изображений. Формат является аппаратно-независимым, поэтому вывод изображений допустим на любых устройствах — от экрана до фотоэкспонирующего устройства. Мощный алгоритм сжатия со средствами управления итоговым разрешением изображения обеспечивает компактность файлов при высоком качестве иллюстраций.

Средством усиления зрительного впечатления и повышения информационной насыщенности изображения служит *цвет*. Ощущение цвета формируется человеческим мозгом в результате анализа светового потока, попадающего на сетчатку глаза от излучающих или отражающих объектов. Цветовые рецепторы (колбочки) подразделяются на три группы, каждая воспринимает только единственный цвет — красный, зеленый или синий. Управление цветом позволяет более точно воспроизводить на экране цвета, которые будут получены при выводе изображения на печать. Все это облегчает работу художников и фотохудожников.

Основы компьютерной графики заключаются в следующем.

Графический режим построения изображений на экране осуществляет *видеоадаптер*, управляющий работой электронной трубки и видеопамятью, в которой запоминается текущее изображение. Адаптер обеспечивает регулярное отображение содержимого видеопамяти на экране дисплея.

Большинство языков программирования имеют свои графические библиотеки (например, Турбо Паскаль — Graph.tpu, Бейсик — графические встроенные команды). Графические редакторы, издательские системы и др. имеют удобный интерфейс для построения всевозможной графики, от простейших рисунков до мультипликационных (анимационных) роликов.

Построение компьютерных изображений требует геометрической грамотности и интуиции, конструкторских и изобразительных навыков. Прежде всего, надо знать следующие понятия и уметь применять свое знание на практике.

Композиция — строение, соотношение и взаимное расположение частей, сосредоточие идейно-творческого начала, позволяющего автору произведения искусства целенаправленно организовывать главное и второстепенное и добиваться максимальной выразительности содержания и формы в их образном единстве. Законы композиции касаются цельности, контрастов. Правила композиции: передача ритма; композиционный центр.

Мера — характеризует общие принципы строения, целостность предмета, лежит в основе ритма, гармонии, ансамбля в архитектуре.

Симметрия — одинаковость в расположении частей чего-нибудь по противоположным сторонам от точки, прямой или плоскости. Симметрия в композиции создается уравновешенностью ее частей по массам, тону, цвету и форме. Симметрия (геометрическая) — свойство геометрических фигур, при котором каждая пара соответственных точек лежит на одном перпендикуляре к данной плоскости по разные стороны и на одинаковом расстоянии от нее.

Пропорция — определенное соотношение частей между собой, соразмерность.

Ритм — равномерное чередование каких-нибудь элементов.

Гармония — соразмерность частей, слияние различных компонентов объекта в единое органичное целое. В истории эстетики гармония — существенная характеристика прекрасного.

Перспектива — искусство изображать на плоскости трехмерное пространство в соответствии с тем кажущимся изменением величины, очертаний, четкости предметов, которое обусловлено степенью отдаленности их от точки наблюдения.

Выделяют два принципа моделирования графических объектов на экране дисплея: *случайный* и *детерминированный*. На их основе строятся технологии изобразительной графики, имеющей три направления: художественное, иллюстративное и демонстрационное.

Объекты художественной графики: различные узоры (самые простые — бордюры, орнаменты, паркеты — разбиение плоскости на непересекающиеся многоугольники), шрифты и другие изображения.

Объекты иллюстративной графики: схемы, эскизы, географические карты, чертежи и др. Создаются сочетанием случайного и упорядоченного в любой пропорции. Могут быть представлены в демонстрационной, динамической форме.

Объекты демонстрационной графики — динамические объекты. В технологии их использования выделяют три способа: рисование/стирание, смену кадров (страниц), динамические образы.

Существуют следующие виды графики.

Деловая графика — отображение данных экономических расчетов (расчетных и статистических) в виде схем, диаграмм, гистограмм, графиков.

Гистограмма — группа столбцов, пропорциональных по высоте определенным числовым значениям.

Круговая гистограмма — сектора круга, углы которых пропорциональны элементам данных.

Круговая диаграмма — представление относительных величин объектов, которым на изображении сопоставляются размеры и расположение кругов в прямоугольной системе координат.

Временная диаграмма — последовательность операций или процессов определенной длительности (измерение динамических процессов).

Линейный график — отображение исходных величин в виде точек, соединенных отрезками прямых линий.

Структурная схема — представление сложных объектов в виде дерева или графа.

Инженерная графика — компьютеризация чертежных и конструкторских работ с помощью систем автоматизации проектных работ САПР (AutoCad, TFlex и др.), представляющих собой аппаратно-программный комплекс, поддерживающий процесс проектирования с использованием специальных средств машинной графики. Используют в архитектуре, строительстве, картографии, медицине, геофизике, разработке моделей одежды, издательском деле, рекламе и др. Например, возможности системы AutoCad следующие:

- развитая система экранных меню;
- высокая точность графической информации;
- разбивка информации (расслоение);
- прочерчивание на экране координатной сетки;
- средство захвата графических объектов;
- мощное редактирование;
- отображение параметров графических характеристик;

- полуавтоматическая и автоматическая установка размеров;
- штриховка;
- работа с блоками.

Научная графика — средства формирования научной документации, имитационного, аналитического, геометрического и координатного моделирования.

Для компьютерного моделирования графического объекта следует выбрать подходящее программное инструментальное средство — графический редактор. Иногда полезно осуществить построение графического образа программированием с использованием графических библиотек.

Существует много графических редакторов — программ для работы с графикой.

Растровые редакторы могут быть двух видов: только для рисования (Paint, Painter) и для обработки растровых изображений (Adobe Photoshop, Photos Tyler, Picture Publisher, Corel Painter, Gimp, Jasc Paint Shop, Pro Premiu, Corel PrintHouse и др.).

Особое место занимает программный пакет *Adobe Photoshop* для обработки растровой графики. Он сегодня является стандартом в компьютерной графике. В нем улучшен контроль над цветом, параметрами изображения, оттенками и прочими деталями. Изначально программа была разработана для редактирования изображений для печати, но она все чаще используется при создании изображений для Всемирной паутины.

Главные элементы управления программы сосредоточены в строке меню, на панели инструментов и панели свойств. Особую группу составляют инструментальные палитры. Функции перечисленных средств рассматриваются в разделе 5.3.2.

Среди программ, предназначенных для создания компьютерной двумерной живописи, самыми популярными считаются *Painter* компании Fractal Design, *FreeHand* компании Macromedia и *Fauve Matisse*. Пакет Painter обладает достаточно широким спектром средств рисования и работы с цветом. Он моделирует различные инструменты (кисти, карандаш, перо, уголь, аэрограф и др.), позволяет имитировать материалы (акварель, масло, тушь), а также добиться эффекта натуральной среды. Последние версии программы FreeHand обладают богатыми средствами редактирования изображений и текста, содержат библиотеку спецэффектов и набор инструментов для работы с цветом, в том числе средства многоцветной градиентной заливки. Программа для создания изображений на платформе Macintosh — пакет для ре-

дактирования растровой живописи и изображений *PixelPaintPro* компании Pixel Resources.

Среди программ компьютерной живописи для графических станций Silicon Graphics (SGI) особое место занимает пакет *StudioPaint 3D* компании Alias Wave Front, который позволяет рисовать различными инструментами (кистями) в режиме реального времени прямо на трехмерных моделях. Пакет работает с неограниченным количеством слоев изображения и предоставляет 30 уровней отмены предыдущего действия (Undo), включает операции цветокоррекции и сплайновые кисти, мазок которых можно редактировать по точкам как сплайновую кривую. StudioPaint 3D поддерживает планшет с чувствительным пером, что дает возможность художнику сделать традиционный эскиз от руки, а затем перенести рисунок в трехмерные пакеты для моделирования или анимации и построить по эскизу трехмерную модель.

К аппаратным средствам получения цифровых растровых оригиналов в основном относятся *сканеры* и *цифровые фотокамеры*. Другие устройства, например цифровые видеокамеры, адаптеры захвата телевизионных кадров, в компьютерной графике играют чаще вспомогательную роль. Для создания изображений «от руки» предназначены *графические планшеты*, на которых рисуют специальным электронным пером.

При создании художественных композиций отдельные фрагменты берут из библиотек (Clip Art).

Особый класс программ — «катализаторы». Они позволяют просматривать графические файлы (множество форматов), создавать на диске альбомы, перемещать и переименовывать файлы, документировать и иллюстрировать издания.

Corel Painter — мощнейший в мире редактор для рисования и создания иллюстраций, предлагает своим пользователям уникальные электронные кисти, а также средства имитации материалов и текстур, полностью повторяющих вид и свойства своих природных аналогов.

GIMP — редактор, в котором вместо привычного окна программы с обширными главным меню и панелью инструментов имеется миниатюрное сосредоточение кнопок. Стоит только открыть какое-либо изображение — документы открываются в независимых окнах, в которых присутствует главное меню. GIMP может использоваться как редактор для рисования и как инструмент для обработки фотографий. Он включает в себя более

200 расширений. Все инструменты рисования гибко настраиваются. Работа с текстом идет с помощью стандартного набора инструментов. Рисунки художественных эмблем делаются с помощью специальных сценариев. Создание анимации — работа с кадрами анимации как с отдельными слоями изображения. Можно преобразовывать изображения: вращать, масштабировать, наклонять и отражать. GIMP позволяет создавать новые инструменты на основе группы фильтров. При работе с экспозицией есть автоматический режим, позволяющий улучшать изображения одним щелчком.

Векторные редакторы: Adobe Illustrator, CorelDraw, Macromedia Freehand. Все эти редакторы работают с одними и теми же векторными графиками, имеют схожие инструменты.

Векторизаторы (трассировщики) — специализированные пакеты преобразования растровых изображений в векторные (например, *Adobe StreamLine*, *Corel Trace*).

Adobe Illustrator — эта программа является общепризнанным мировым лидером среди средств работы с векторной графикой. Ее достоинство: вместе с Photoshop и настольной издательской системой PageMaker образует трио для создания изображений, достаточных для верстки полиграфических изданий. Эти приложения выполнены в едином стиле, используются похожий интерфейс и инструменты, позволяющие применять одинаковые приемы и навыки, экспортируют и импортируют созданные объекты между собой.

Macromedia Freehand с простым и дружественным интерфейсом служит удобным инструментом работы для начинающих. Программа отличается небольшим размером и хорошим быстродействием. Нетребовательность к аппаратным ресурсам позволяет работать на компьютерах среднего уровня. Инструментальные средства программы достаточны для разработки сложных документов. Пакет специально адаптирован для совместной работы с программой компьютерной верстки *QyarkXPress*.

CorelDraw исторически, особенно в России, считается основным пакетом создания и обработки векторной графики на платформе Windows. К его преимуществам относятся развитая система управления и обширные средства настройки параметров инструментов. По возможностям создания самых сложных художественных композиций CorelDraw превосходит конкурентов. Однако интерфейс программы считается непростым для освоения. CorelDraw — это развитая система управления, богатые

средства настройки инструментов, наиболее сложные композиции, близкие к художественным произведениям.

Трассировщик *Adobe StreamLine* по праву занимает ведущее место в своем классе программ. Имеются более мощные пакеты, ориентированные на обработку чертежей, но они очень требовательны к аппаратным ресурсам, да и по стоимости много дороже. StreamLine позволяет проводить тонкую настройку параметров векторизации, что улучшает ее точность. Более всего векторизация удобна для преобразования чертежей, черно-белых рисунков и другой простой графики без полутонов. Полутоновые и цветные изображения обрабатываются хуже и для приближения к оригиналу требуют значительной доработки.

Контрольные вопросы

1. Каковы виды компьютерной графики?
2. Что такое растровая графика?
3. Что такое векторная графика?
4. Что такое фрактальная графика?
5. Какие вы знаете форматы файлов для хранения изображений?
6. Для чего используют цвет в компьютерной графике?
7. Что такое композиция, мера, симметрия?
8. Что такое пропорция, ритм, гармония, перспектива?
9. Что такое деловая, инженерная и научная графика?
10. Какие вы знаете растровые редакторы?
11. Какие вы знаете векторные редакторы?

5.3.1. Создание сложного рисунка в векторном редакторе Word и растровом редакторе Paint

Для создания рисунка с помощью встроенного графического редактора Word следует выполнить следующие действия:

- создать новый документ, развернуть окно документа;
- вывести на экран непечатаемые символы;
- ввести текст;

- после, например, первого абзаца текста вставить несколько пустых абзацев, используя принудительный перевод строки с помощью комбинации клавиш **Shift+Enter**;
- создать с помощью инструментов панели **Рисование** рисунок, используя фигуры прямоугольника и овала, стрелки, надписи, цвета, действия; сохранить файл;
- объединить элементы рисунка можно, выделив его (нажав клавишу **Shift** и щелкая по элементам мышью или с помощью кнопки **Выбор объектов** на панели **Рисование**);
- далее рисунок следует сгруппировать с помощью кнопки **Действия** и команды **Группировать**;
- чтобы убедиться, что все элементы рисунка сгруппированы, нужно его переместить;
- для привязки рисунка к абзацу следует рисунок выделить, а символ *якоря* перетащить мышью к нужному абзацу; введя команды **Формат** → **Объект** → **Рисунок/Автофигуры**, на вкладке **Положение** установить соответствующий переключатель.

Создать в документе с многослойным рисунком объект **WordArt** и отредактировать его с помощью панели **WordArt** и кнопок на панели **Рисование**.

Создать в документе любую плоскую автофигуру, изменить ее размер и скопировать (перетащив ее мышью при нажатой клавише **Ctrl**), используя кнопку **Автофигуры**.

Для запуска программы Paint следует последовательно выбрать пункты меню **Пуск** → **Программы** → **Стандартные** → **Графический редактор Paint**. На экране появится окно Paint с чистым рабочим полем для рисования, панелью инструментов, палитрой цветов и указателем координат курсора. Размеры рабочего поля можно менять (растянуть по горизонтали, вертикали, диагонали; свернуть, развернуть, восстановить).

Палитра цветов Paint позволяет работать сразу с двумя цветами: цветом переднего плана и цветом фона. Цветом переднего плана (основным цветом) обычно закрашиваются основная часть рисунка и текстовые символы. Два смещенных относительно друг друга прямоугольника в левой части палитры указывают текущие цвета: верхний — основной, нижний — цвет фона. Изменение цветов осуществляется следующим образом: основного — щелчком левой кнопки мыши по любому из цветов палитры, цвета фона — щелчком правой кнопки.

Панель инструментов. Каждый инструмент обозначен небольшой кнопкой (пиктограммой). Щелчком мыши по кнопке можно активизировать выбранный инструмент.

Выделение произвольной области. Указатель мыши принимает вид крестика. Нажав левую кнопку мыши, можно перемещать его в области рисунка, при этом выделяя произвольную область рабочего поля. Если клавишу отпустить, то нарисованная линия превратится в пунктирный прямоугольник. Эту выделенную область можно преобразовывать с помощью команд меню **Правка** (поместить в буфер обмена, удалить или переместить в другую часть рисунка). Выделить сразу весь прямоугольник можно с помощью инструмента **Выделение**.

Ластик/Цветной ластик. Указатель мыши превращается в квадратный ластик. Нажав левую кнопку мыши и перемещая ластик, можно закрасить поверхность цветом фона. Если цвет фона белый, то ластик просто стирает участки рисунка. Нажав правую кнопку мыши, можно пользоваться цветным ластиком. При этом стираемый цвет выбирается как основной, а цвет фона — белым. Размер ластика выбирают с помощью дополнительного раздела в левой нижней части панели инструментов.

Заливка. Указатель превращается в копию кнопки **Заливка**. Если кончик «льющейся струи краски» расположить в замкнутой области и щелкнуть левой кнопкой мыши, то область закрасится основным цветом, если правой — цветом фона.

Выбор цветов. Указатель мыши принимает вид пипетки. Следует указать пипеткой на объект, цвет которого нужно скопировать, и щелкнуть левой кнопкой мыши. Произойдет копирование выбранного цвета в основной цвет (при этом в смещенных друг относительно друга прямоугольниках верхний окрасится скопированным цветом). Если необходимо скопировать цвет фона или ластика, то следует щелкнуть правой кнопкой мыши.

Масштаб. С помощью этого инструмента можно увеличивать или уменьшать изображение в несколько раз.

Карандаш. Курсор принимает вид карандаша. Нажав левую кнопку мыши, рисуют произвольные фигуры линиями толщиной в один пиксель, в основном цвете. Нажав правую кнопку мыши, можно рисовать в цвете фона.

Кисть. Указатель мыши принимает вид крестика. Им рисуют фигуры так же, как и карандашом. Можно выбрать форму кисти, ее толщину, форму мазка. Форма кисти выбирается с помо-

щью дополнительного раздела, который появляется в нижней части панели инструментов при активации кнопки **Кисть**.

Распылитель. Указатель мыши превращается в «струю аэрозольного баллончика». Нажав левую кнопку мыши и перемещая указатель по экрану, окрашивают поверхность струями (мазками) в основном цвете.

Ширину мазка выбирают с помощью дополнительного раздела, который появляется в нижней части панели инструментов при активации кнопки **Распылитель**. Нажав правую кнопку мыши, можно «распылять» цвет фона.

Надпись. Позволяет вводить текст в поле рисунка. Выбрав этот инструмент и нажав левую кнопку мыши, рисуют текстовую рамку. Щелкнув мышью внутри рамки, набирают текст. Открыв меню **Вид** и установив флажок **Панель атрибутов текста**, выбирают нужный шрифт, его размер и стиль. Цвет текста устанавливают с помощью дополнительного раздела, который появляется в нижней части панели инструментов при активации кнопки **Надпись**.

Линия. Указатель мыши принимает вид крестика. Им проводят прямые линии.

Кривая. Указателем мыши рисуют прямую линию, а затем, потянув в любом месте в нужном направлении, можно изогнуть ее в дугу разного вида.

Геометрические фигуры. Набор инструментов: прямоугольник, многоугольник, эллипс, скругленный прямоугольник. Перемещая курсор, рисуют названные выше фигуры. Они могут быть неокрашенными/иметь основной цвет. Окрашивание внутри контура или без наружного контура достигается выбором соответствующей фигуры в дополнительном разделе, который появляется в нижней части панели инструментов при активации кнопки этой фигуры. Нажав правую кнопку мыши, можно «распылять» цвет фона.

Дополнительные возможности редактора:

1. Меню **Рисунок**. Команда **Отразить/повернуть...** позволяет отразить картинку (слева направо или сверху вниз) и повернуть ее на угол 90, 180, 270°. Команда **Растянуть/наклонить...** позволяет изменить пропорции изображения.

2. Меню **Правка**. Команда **Очистить выделение** позволяет очистить выделенный фрагмент. Можно отменить не одну, а три последовательно выполненные операции.

Для рисования сложного рисунка в векторном редакторе Word и растровом редакторе Paint нужно выполнить следующие действия:

1. Запускаем программу MS Word. С помощью команд **Вид → Панели инструментов → Рисование** выводим на экран панель **Рисование**. Изучаем все инструменты панели **Рисование**. Для создания рисунков выполняем следующие действия:

- создаем документ с именем Изобр.doc, открываем окно документа;
- выводим на экран непечатаемые символы;
- создаем с помощью инструментов панели **Рисование** рисунок и сохраняем файл;
- объединяем все элементы рисунка;
- создаем в документе объект WordArt, редактируем его;
- группируем рисунок;

На экране получаем следующий рисунок (рис. 5.19).

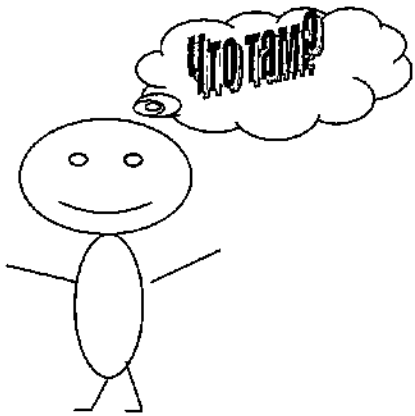


Рис. 5.19. Рисунок, созданный в программе Word

2. Запускаем программу Paint. Изучаем палитру цветов и панель инструментов. Для создания изображения выполняем следующие действия: в рабочем поле создаем рисунок, делаем надпись, окрашиваем рисунок и надпись, редактируем созданное изображение (рис. 5.20).

3. Сохраняем изображение в графическом файле с расширением .bmp и копируем его в ранее созданный с помощью текстового редактора файл, группируем. На экране получаем сложный многослойный рисунок с надписями (рис. 5.21).

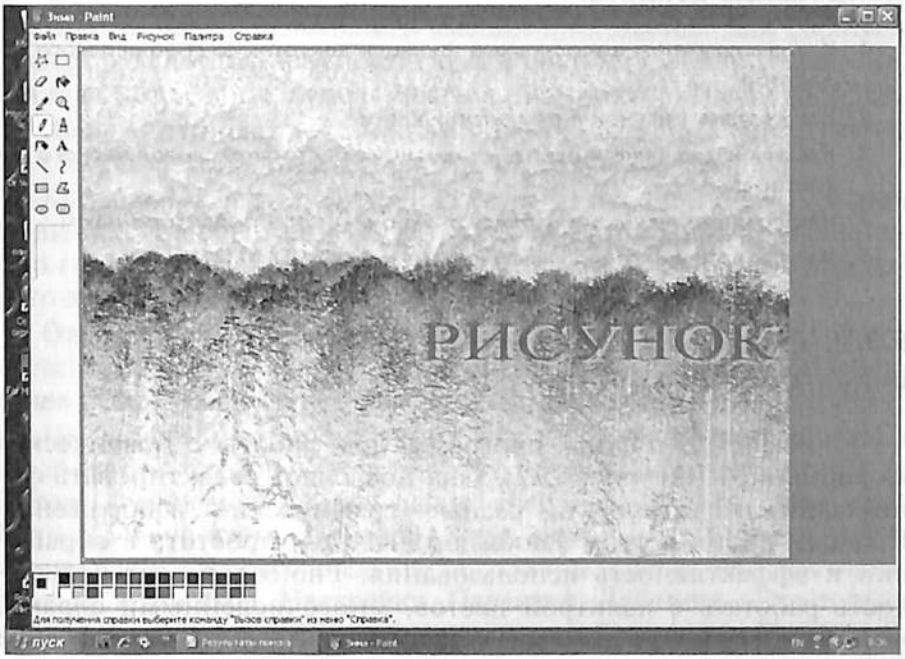


Рис. 5.20. Рисунок, созданный в программе Paint

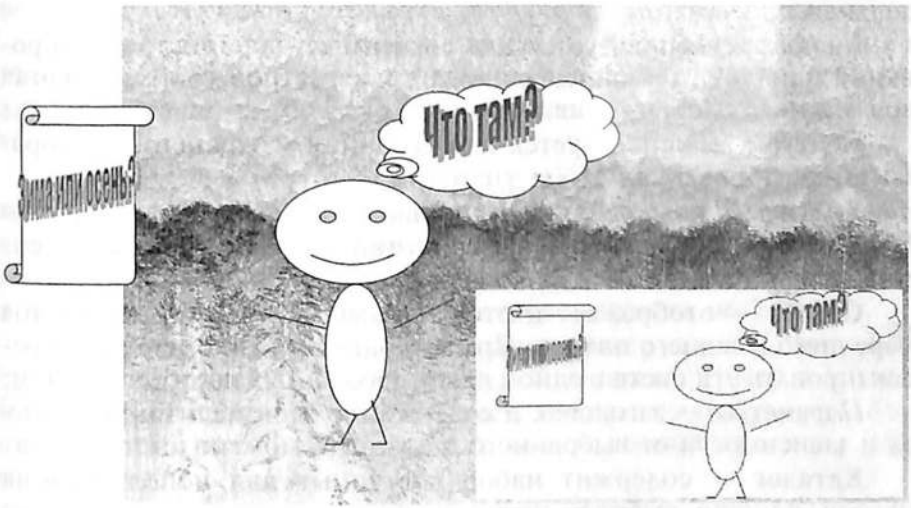


Рис. 5.21. Окончательный вид рисунка

Контрольные вопросы

1. Каковы назначение и основные функции векторного графического редактора?
2. Как создать рисунок в редакторе Word?
3. Каковы назначение и основные функции растрового графического редактора?
4. Как создать рисунок и надпись в рабочем поле редактора Paint?
5. Как создать многослойный рисунок?

5.3.2. Рисование сложного рисунка в растровом редакторе Photoshop

Photoshop — мощная программа для работы с графической информацией (см. рис. 5.22). Она позволяет редактировать фотографии, моделировать разные графические изображения. Главным достоинством Photoshop является простота в обращении и эффективность использования. Photoshop дает возможность работать с палитрой цветов, многочисленными слоями, каналами, масками, цветами, фильтрами, изменять изображение и его разрешение, выполнять печать, сканирование и т. д.

Палитры Photoshop. Слои — предназначена для создания, копирования, наложения слоев-масок, помогает изменению отдельных слоев.

Каналы — используется для создания, удаления, редактирования каналов, для определения их параметров, формирования совместного изображения.

Контуры — используется для создания, сохранения и обработки контуров.

Кисти — позволяет регулировать размер кисти, причем Photoshop сохраняет все характеристики для каждого инструмента в отдельности.

Синтез — отображает цветовые значения различных цветов переднего и заднего планов. При помощи ползунков можно отредактировать эти цвета в одной из предложенных цветовых систем.

Параметры — заголовок и содержимое этой палитры меняются в зависимости от выбранного на данный момент инструмента.

Каталог — содержит набор доступных для использования цветов. Можно выбрать цвета переднего и заднего планов из имеющихся вариантов.

Дизайн — представляет собой планшет, который можно использовать для произвольного смешивания цветов. Для работы в палитре можно пользоваться любыми рисующими инструментами. Для просмотра могут использоваться инструменты **Масштаб, Рука**.

Инфо — отображает информацию об активном инструменте и о цветовых координатах в активной точке курсора.

Команды — если палитра открыта на экране, щелчок мыши по любой из содержащихся в ней команд приводит к выполнению соответствующей операции или к открытию соответствующего диалогового окна.

Окно Photoshop. В строку меню, которая находится сверху, входят следующие пункты меню: **Файл, Редактирование, Изображение, Слой, Выделение, Фильтр, Вид, Окно**.

На панели инструментов (см. рис. 5.22), которая находится в левой части экрана, имеются следующие инструменты: **Рука, Ластик, Увеличение, Уменьшение, Вырезать, Залить, Карандаш, Кисть, Подбор цвета, Пипетка, Выделение** и т. д. В правой части экрана располагаются следующие палитры: **Инструмент, История, Каналы, Навигатор, Настройки, Параграф, Действия**... Это только часть палитр, которые можно вывести в рабочее окно Photoshop.

Чтобы вывести дополнительные палитры, следует в меню **Окно** выбрать то окно, которое требуется. Например: **Окно → Кисти**. Используя эти палитры и зная их назначение, можно грамотно редактировать, создавать графическую информацию.

Рисование — самая примитивная и простейшая функция Photoshop. Оно производится при помощи инструментов **Кисть** либо **Карандаш**, которые можно выбрать на панели инструментов слева.

Клонирование — этот инструмент также находится на панели инструментов слева. После нажатия на кнопку **Клонирование** следует выбрать место, с которого будет производиться клонирование, оно выбирается путем нажатия клавиши **Alt** и щелчка мышью по тому месту, с которого надо клонировать объект.

Заливка производится одноименным инструментом, который также находится на панели инструментов слева. После нажатия на кнопку **Заливка** выбирается цвет, которым надо залить объект, а затем, нажав левую кнопку мыши, нужно залить то или иное место.

Сложение слоев выполняется при помощи следующих команд: **Слой → Выполнить слияние**. При выполнении этих команд

те слои, которые находятся в данный момент на главном документе, сложатся в один и получится целостный документ.

Ввод текста на экран производится при помощи инструмента **Текст**, расположенного на панели инструментов слева. Нажимаем на кнопку с буквой Т, а затем выбираем место, где требуется ввести текст, и вводим его с клавиатуры.

Сохранение документа производится при помощи команд меню **Файл → Сохранить**, но перед этим следует поставить такое разрешение изображения, которое требуется (**Изображение → Размер изображения**). Следует указать размер документа и его разрешение (пикс.\дюйм). Затем сохранить документ.

Импорт документа производится при помощи команд **Файл → Импорт**. При помощи этой команды изображение можно импортировать в приложения WIA, PDF и т. д. Экспорт документа производится при помощи команд **Файл → Экспорт**. Можно экспортировать документ в такие программы, как Illustrator, Zoomview.

Распечатка документа производится при помощи команд **Файл → Печать** с предварительным просмотром. Выбирается размер документа, количество копий, отступы, высота, ширина, установки страницы.

Для рисования сложного рисунка в растровом редакторе Photoshop нужно выполнить следующие действия:

1. Выполняем вставку скриншота из игры в документ Photoshop при помощи клавиши **Print Screen** (рис. 5.22). Далее указываем размер документа (в данном случае —768 × 1024 пикселей). Для того чтобы изображение переместилось в максимально идентичном виде, использовался встроенный фильтр, диалоговое окно которого появляется автоматически (рис. 5.23).

После вставки изображения его можно сразу редактировать и изменять на экране. Для этого использовались функции **Сжижение**, **Копирование**, **Клонирование**, **Вставка**, **Вставка примечания**, **Заливка**, **Сложение слоев**, **Совмещение 2 документов**, **Рисование**, **Лечащая кисть**, ввод текста непосредственно на экран и т. д.

2. Для совмещения двух изображений (или их частей):

- открываем изображение, которое нужно вставить;
- выбираем команды **Выделение → Выделить все**;
- нажимаем кнопку **Копировать**, переходим в главный (основной) документ и нажимаем кнопку **Вставить** (рис. 5.24). Получаем на фоне главного документа многослойный документ или часть документа, которую мы выделили (рис. 5.25).

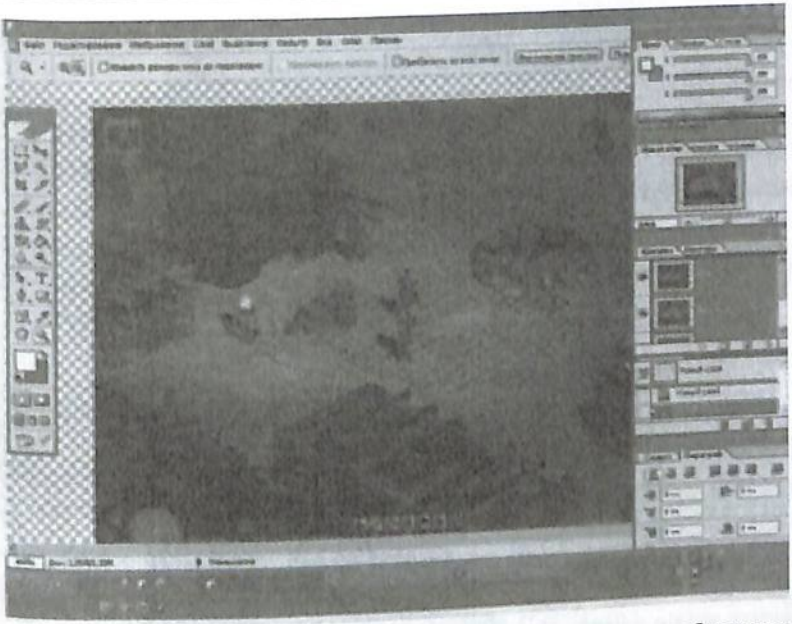


Рис. 5.22. Окно растрового редактора Photoshop. Исходное изображение

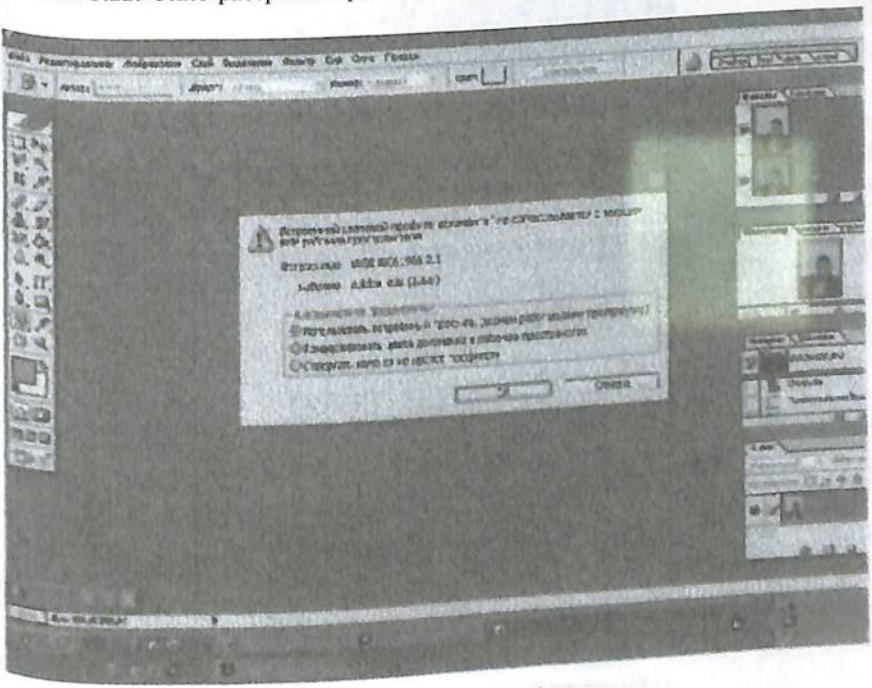


Рис. 5.23. Встроенный фильтр

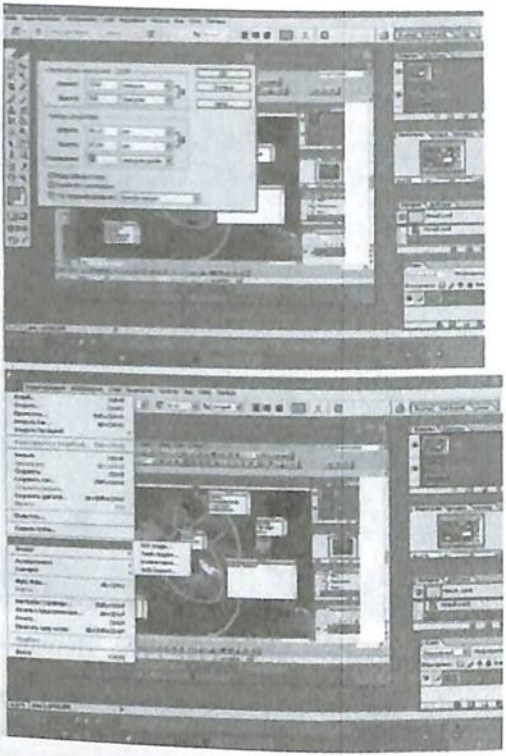


Рис. 5.24. Получение совмещенного изображения

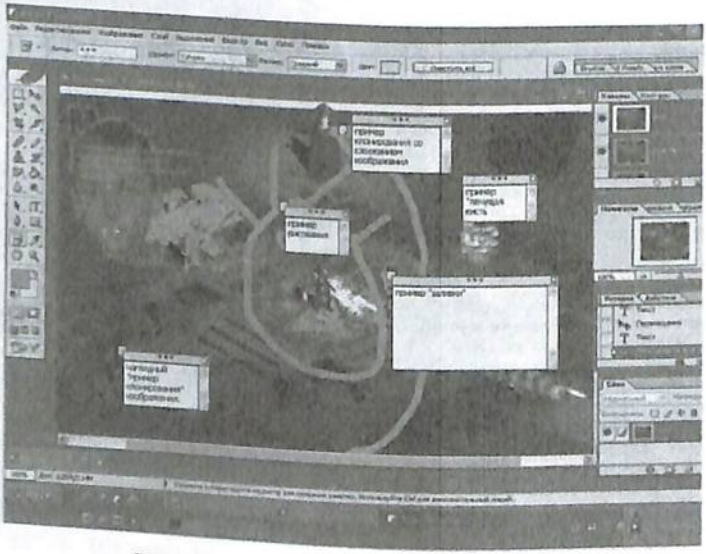


Рис. 5.25. Многослойное изображение

Контрольные вопросы

1. Каковы назначение и основные функции растрового графического редактора Photoshop?
2. Какие палитры имеются в редакторе Photoshop?
3. Как создать рисунок и надпись в рабочем поле редактора Photoshop?
4. Как создать многослойный рисунок в редакторе Photoshop?
5. Как осуществить вставку рисунка в файл и сохранить изображение в графическом файле?

5.3.3. Система MS PowerPoint

PowerPoint — программная среда, предназначенная для подготовки презентаций (от англ. *Presentation* — представление). PowerPoint — многооконное приложение Windows.

Процесс создания презентации начинается с отбора и подготовки отдельных, сменяющих при демонстрации друг друга слайдов. *Слайд* — фрагмент презентации, в пределах которого производится работа над ее объектами. Слайд может включать текст, рисунки, клип, звук и др. В совокупности презентация — набор слайдов (слайд-фильм) на определенную тему, который хранится в файле специального формата с расширением .ppt.

Перед созданием презентации нужно составить ее *проект*, т. е. решить, из каких объектов будет состоять каждый слайд. И в соответствии с количеством и видом объектов наметить этапы их создания.

Для разработки проекта презентации нужно выполнить следующие этапы:

- создание слайдов;
- создание фона;
- создание текста;
- вставка рисунков;
- настройка анимации рисунков и текста;
- запуск и наладка презентации.

Запуск PowerPoint можно осуществить из главного меню Windows, выбрав программу PowerPoint или нажав соответствующую кнопку пиктографического меню MS Office. Предлагаемая презентация создана с помощью приложения PowerPoint 2007, работающего под управлением ОС Windows Vista.

После запуска PowerPoint появляется окно приложения с заготовкой первого слайда и меню. Для создания новой презентации следует открыть вкладку меню **Главная**. На открывшейся вкладке щелкаем мышью по кнопке **Создать слайд** — появляется диалоговое окно, в котором располагаются шаблоны слайдов (рис. 5.26).

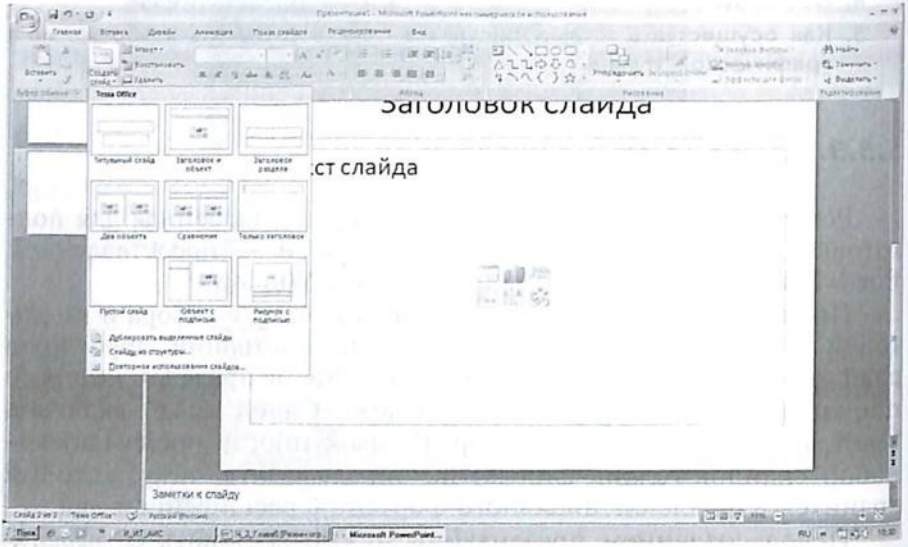


Рис. 5.26. Диалоговое окно с шаблонами слайдов

Необходимо выбрать из них подходящую форму слайда и далее формировать его по своему усмотрению.

Раскрывая вкладки меню **Главная**, **Вставка**, **Дизайн**, **Анимация**, **Показ слайдов**, **Рецензирование**, **Вид**, нужно установить необходимые для работы инструменты, изучить их функции.

Инструменты вкладки **Главная** позволяют открывать, записывать, создавать, отправлять на печать презентации, копировать и вырезать объекты, изменять шрифты (стиль, размер, вид), выравнивать текст, работать с абзацами, рисовать и редактировать.

Инструменты вкладки **Вставка** позволяют вставлять таблицы, рисунки, клипы, фотографии, диаграммы, связи по гиперссылкам, текст, клипы мультимедиа.

Инструменты вкладки **Дизайн** выполняют функции установки параметров страницы, ориентации слайда, выбора тем, установки цветов и эффектов, стилей фона.

Инструменты вкладки **Анимация** дают возможность просмотра и настройки анимации, установки стандартных звуков и смены слайдов по щелчку или автоматически по времени.

Инструменты вкладки **Показ слайдов** определяют порядок показа слайдов, настройку демонстрации, запись речевого сопровождения, настройку времени, режим доклада и др.

Инструменты вкладки **Рецензирование** позволяют следить за орфографией, осуществлять поиск в справочниках, создание примечаний и др.

Инструменты вкладки **Вид** выполняют функции сортировки слайдов, установки масштаба, работы с окнами, создания макросов и др.

Например, чтобы вставить рисунки в слайды, можно использовать команды **Вставка** → **Рисунок**. Открывается диалоговое окно **Вставка рисунка**, где показаны файлы с рисунками или отдельные фотографии. Выбрав фотографию, ее можно скопировать в слайд (рис. 5.27).

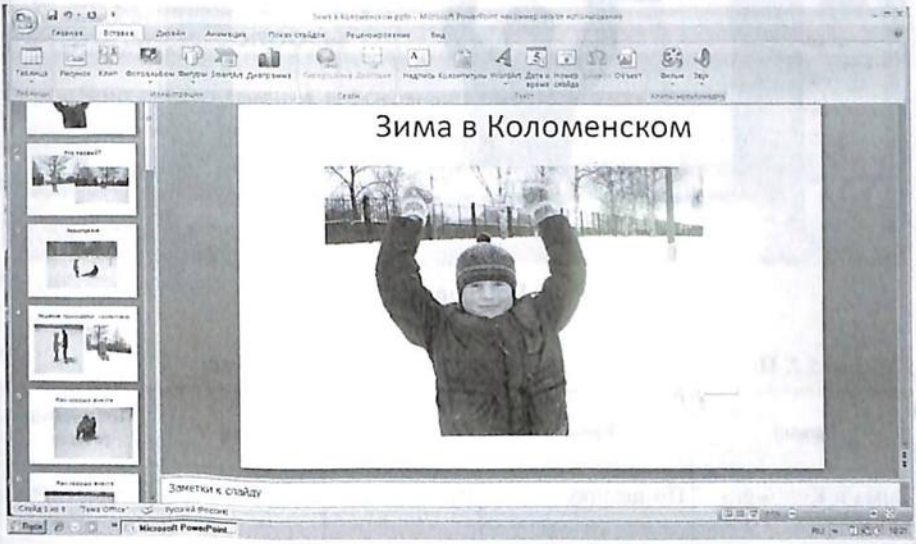


Рис. 5.27. Вставка фотографий

Можно выбрать необходимый рисунок из предлагаемой галереи Clip Gallery (или скопировать в буфер обмена из файла) и затем вставить его в слайд.

Надписи можно создавать с помощью команд **Вставка** → **Надпись** или вводя текст в предлагаемые шаблоны.

Для создания эффектов анимации следует последовательно выбрать команды **Анимация** → **Анимация** → **Настройка анимации** → **Добавить эффект**. Выделяем на слайде объект, которому будет задан эффект. В раскрывающемся списке устанавливаем процесс (например, **Вход**) и эффект (например, **Вращение**) — рис. 5.28. Лучше заранее расписать все устанавливаемые эффекты. Пример представлен в табл. 5.2.

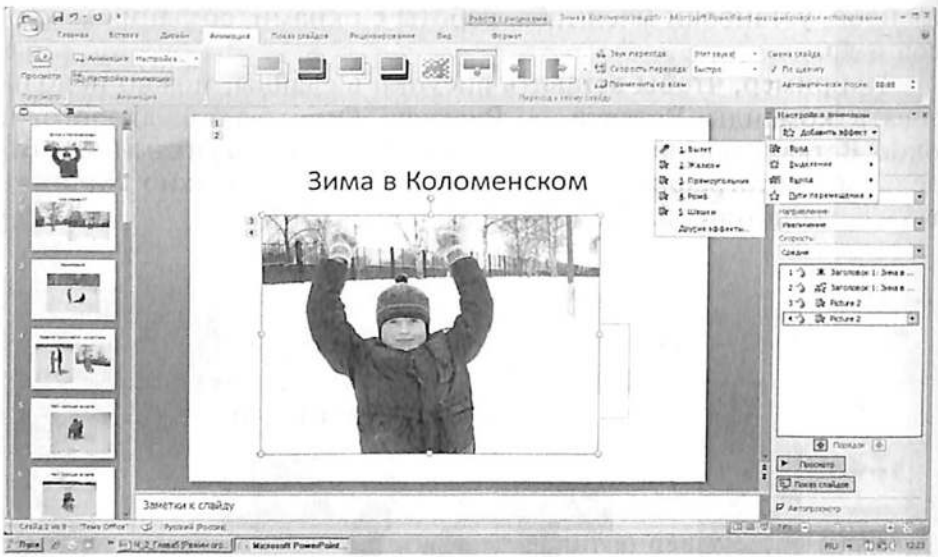


Рис. 5.28. Настройка анимации

Таблица 5.2. **Пример плана назначения эффектов для презентации**

Объект	Время	Эффекты	Звук	Появление текста
Зима в Коломенском	По щелчку	Вращение	Laser.wav	Все вместе
Фотография на слайде 1	Автоматически	Выползание	Laser.wav	—
Текст на слайде 2	Автоматически	Часовая стрелка	Remider.wav	По буквам

Окончание табл. 5.2

Объект	Время	Эффекты	Звук	Появление текста
Фотография на слайде 2	Автоматически	Панорама	Камера	—
Фотография на слайде 3	Автоматически	Круговой	Телефон	—
Текст на слайдах 3—6	Автоматически	Выползание слева	Laser.wav	По буквам
Фотография на слайдах 7—8	Автоматически	Выползание	Гонки	—

Для демонстрации презентации выбираем из меню пункт **Показ слайдов** и запускаем слайд-шоу либо с начального слайда, либо с произвольно выбранного.

Контрольные вопросы

1. Каковы назначение и основные функции среды PowerPoint?
2. Что такое слайд и как его создать?
3. Какие действия можно выполнять с помощью используемых для создания презентации диалоговых окон?
4. Как настроить анимацию текста и рисунка?
5. Как озвучить эффект анимации?

Глава 6
ОСНОВЫ БАЗ ДАННЫХ И ЗНАНИЙ. СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ (СУБД)

Человечество на протяжении тысячелетий накапливало сведения, знания об окружающем нас мире. Для того чтобы знания не исчезали, а передавались из поколения в поколение, их передача осуществлялась через предания, потом через книги, телеграф, телевидение и др., т. е. через средства, позволяющие накапливать сведения об объектах окружающего мира. Сведения об объекте или отношениях объектов, выраженные в знаковой форме, образуют данные. Они могут быть восприняты человеком или техническим устройством и соответствующим образом объяснены (интерпретированы). Данные и информация — понятия взаимосвязанные, но не тождественные. *Данные* представляют собой зарегистрированные сигналы и являются составной частью информации. В ходе информационных процессов данные преобразуются из одного вида в другой с помощью методов (наблюдение, сбор, формализация, фильтрация, сортировка, архивация, транспортировка, преобразование и др.).

Информация — это продукт взаимодействия данных и адекватных им методов.

Знания — теоретическая и практическая информация, обладающая свойствами обучения и производства новых знаний (см. также раздел 4.1).

В те годы, когда формировалось понятие БД, в них действительно хранились только данные. Однако сегодня большинство БД позволяют размещать в своих структурах не только данные, но и методы (т. е. программный код), с помощью которых происходит взаимодействие с потребителем или с другими программно-аппаратными комплексами. Таким образом, в современных БД хранятся не только данные, но и информация.

Возможность перевода данных из одной знаковой системы в другую позволяет описывать предметную ситуацию в различных

системах знаков, ориентируясь на пользователя. Эту информацию необходимо структурировать. *Структурирование данных* — это процесс группировки данных по определенным параметрам. Информацию нужно организовать таким образом, чтобы пользователь мог ею воспользоваться. Под *визуализацией информации* базы понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройство вывода или передача по каналам связи.

В конечном счете пришли к построению баз и банков данных, а также знаний.

База данных (*БД*) — именованная совокупность организованных данных, предназначенная для хранения информации и отображающая состояние объектов и их отношений в определенной предметной области.

База знаний (*БЗ*) — именованная совокупность организованных данных и знаний в определенной предметной области и логические правила манипулирования ими для получения необходимых, в том числе новых, знаний.

Знания в БЗ должны быть представлены в такой форме, чтобы они могли быть легко обработаны в ЭВМ. Алгоритм обработки знаний заранее неизвестен и строится по ходу решения задачи на основании эвристических правил. Эвристики (правила), по которым решаются задачи, хранятся также в БЗ.

Для формирования БЗ используют три класса способов приобретения знаний:

- диалог эксперта с инженером по знаниям;
- автоматическая генерация знаний;
- построение индивидуальной модели исследования предметной области конкретным экспертом.

Кроме того, необходимы знания в области математической логики и методов представления знаний, знания возможностей ЭВМ, языков и систем программирования. Для разработки БЗ нужны специалисты, обладающие этими знаниями и исполняющие роль посредников между экспертами в предметной области и системами.

Интеграция данных в базах подразумевает совместное использование данных для решения различных задач. Однако это требует централизованного управления, которое называется администрированием данных. Коллектив специалистов, обслуживающий большие БД, включает администратора, аналитиков, системных и прикладных программистов.

Администратор — специалист, имеющий представление об информационных потребностях конечных пользователей и отвечающий за определение, загрузку, защиту и эффективность БД.

Банк данных (*БнД*) — система специально организованных данных, программных, языковых, организационных и технических средств, предназначенных для централизованного накопления и коллективного многоцелевого использования данных.

Пример — БД крупного банка. В ней есть все необходимые сведения о клиентах, об их адресах, кредитной истории, состоянии расчетных счетов, финансовых операциях и др. Доступ к этой базе имеется у достаточно большого количества сотрудников банка, но среди них вряд ли найдется такое лицо, которое имеет доступ *ко всей БД полностью* и при этом способно единолично вносить в нее произвольные изменения. Кроме данных, база содержит *методы и средства*, позволяющие каждому из сотрудников оперировать только с теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными определенным сотрудникам, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и редактирование данных.

БД могут содержать различные объекты, и строятся они по разным моделям.

Простейший некомпьютерный вариант базы данных — деловой ежедневник, в котором каждому календарному дню выделено по странице. Даже если в нем не записано ни строки, он не перестает быть ежедневником, поскольку имеет структуру, четко отличающую его от записных книжек, рабочих тетрадей и пр.

Иерархическая модель — представляет собой совокупность элементов, расположенных в порядке их подчинения от общего к частному и образующих перевернутое дерево (граф).

Сетевая модель — похожа на иерархическую, но порядок взаимодействия элементов иной. В сетевой модели принята свободная связь между элементами разных уровней.

Реляционная модель — в простейшем случае представляет собой двумерный массив или двумерную таблицу, а при создании более сложных информационных моделей — совокупность взаимосвязанных таблиц.

Чтобы обеспечить быстроту и качество поиска данных в БД, процесс ее создания должен быть автоматизирован. Компьютерную модель БД можно создать тремя способами.

1. С помощью языков программирования. Этот способ применяется для создания уникальных БД опытными программистами.

2. С помощью прикладной среды программирования (Visual Basic, Turbo Pascal и др.). Способ требует некоторой обученности работе в программных средах и навыков программирования. С его помощью можно создать БД, требующие каких-то индивидуальных особенностей построения. Создание такой базы под силу только опытным пользователям.

3. С помощью специальных программных средств, которые называются СУБД. Работа с такими системами требует навыков работы с компьютером и может быть освоена пользователями в достаточно короткие сроки. СУБД — совокупность методов, языковых и программных средств, предназначенных для создания, ведения и использования БД многими пользователями. СУБД позволяют создавать и хранить большие массивы данных и манипулировать ими. В настоящее время существует множество СУБД. Например, Oracle, SQL Server, Access, FoxPro, Clipper и др. Каждая из этих систем обладает своими достоинствами и недостатками (табл. 6.1).

Из таблицы можно заключить, что наиболее эффективными для производственных целей являются СУБД Oracle 8, FoxPro, SQL Server, для учебных и офисных — Access.

Рассмотрим элементы реляционной БД. Если в базе нет никаких данных (*пустая база*), то это все равно БД. Хотя данных в БД нет, но информация в ней все-таки есть — это *структура базы*. Она определяет методы занесения данных и хранения их в базе.

Простейшая реляционная база содержит одну таблицу. Соответственно, структура простейшей БД равна структуре ее таблицы. Структуру двумерной таблицы образуют столбцы и строки. Их аналогами в структуре простейшей базы данных являются *поля* и *записи* (рис. 6.1).

	Поле 1	Поле 2	...	Поле *n*
Запись	**Фамилия**	**Имя**	...	**Телефон**
Запись 1	Иванов	Кирилл	...	120-19-50
Запись 2	Петров	Мефодий	...	120-19-75
...	...	...	...	...
Запись *n*	Коптев	Владимир	...	120-19-80

Рис. 6.1. Структура таблицы реляционной БД

Таблица 6.1. **Сравнительные характеристики различных СУБД**

Характеристики СУБД	Oracle 8	SQL Server	Access	FoxPro	Clipper
Скорость работы	Эталон высокопроизводительной реляционной СУБД	Обладает развитыми средствами обработки данных и хорошим быстродействием	Благодаря технологии клиент/сервер снижается объем передаваемых по сети данных, что экономит время пользователей	Несмотря на принцип интеграции, скорость работы программы самая быстрая среди СУБД	Скорость работы программ ниже, чем у FoxPro
Возможность работы с БД других форматов и интеграция с Windows и другими ОС	Возможность согласовывать между собой около 30 вариантов ОС, возможность работать с большими корпоративными приложениями. Отлично работает на UNIX-платформе	Представляет собой клиент/серверную систему управления реляционной БД, тесно интегрированную с Windows NT	Адаптирована к Windows, имеет характерный для всех Windows-приложений удобный графический интерфейс	Адаптирована к Windows, устанавливается на различные платформы	Система замещаемых драйверов БД позволяет применять Clipper к БД «чужих» форматов, адаптирована к Windows
Защита информации и способность избегать конфликтов	Свойство деление данных помогает свести к минимуму возможность возникновения каких-либо проблем. Для защиты системы от несанкционированного доступа используются средства, позволяющие автоматически определять стратегию безопасности при работе с паролями	Устанавливает соединения только с теми пользователями, которых может аутентифицировать или которые уже аутентифицированы средствами Windows NT	Уровень защиты ниже, нежели у других СУБД	Наличие развитой системы обработки ошибок	Наличие развитой системы обработки ошибок

Окончание табл. 6.1

Характеристики СУБД	Oracle 8	SQL Server	Access	FoxPro	Clipper
Удобство использования	Для обеспечения нормальной работы множества пользователей применяется архитектура обработки данных, наличие средств, помогающих сделать процессы резервного копирования и восстановления информации более автоматизированными и быстрыми	Простота администрирования, наличие средств удаленного доступа	Простота в изучении и эксплуатации, снабжена обширными средствами создания отчетов различной степени сложности на основе таблиц различных форматов	Ориентирована на широкий класс пользователей, имеет развитую систему меню и окон, гибкие средства редактирования и просмотра БД, формирование на планшете образа созданного продукта	Пользователь может разрабатывать собственные функции, команды, изменять имеющиеся команды, функции могут быть написаны на языке Си или ассемблере
Общие возможности	Выполнение SQL-запросов, поддержка транзакций			Первоначально имели один и тот же входной язык — DBase	
	—	Для обмена запросами между клиентом и сервером, язык Transact-SQL	—	—	—
Индивидуальные достоинства	Обладает множеством достоинств, в том числе возможностью согласовывать между собой около 30 вариантов ОС	Поддержка огромных БД, многопроцессорность и работа в кластере	Удобство и легкость создания приложений на ее основе, хорошая интеграция с другими СУБД	Обладает всеми достоинствами интегрированной среды и, в частности, очень удобна для обучения, считается самой быстродействующей	Открытая архитектура: пользователь может разрабатывать собственные функции, команды, изменять имеющиеся команды
Соотношение цены и возможностей	Самая мощная, но и самая дорогостоящая	Мощная, средняя цена	Средние возможности, недорогая	Мощная, разумная цена	Мощная, средняя цена

Поле — простейший объект БД, предназначенный для хранения одного параметра реального объекта или процесса.

Запись — совокупность логически связанных полей, характеризующих типичные свойства реального объекта.

Ключ — поле, которое однозначно определяет соответствующую запись. Если записей нет, значит, структура БД образована только набором полей. При изменении структуры базовой таблицы изменяется структура базы данных и, естественно, получается новая база данных.

С БД работают две категории исполнителей. Первая категория — *проектировщики*. Их задача состоит в разработке структуры таблиц базы данных и согласовании ее с заказчиком. Кроме таблиц проектировщики разрабатывают и другие объекты БД, предназначенные, с одной стороны, для автоматизации, а с другой стороны, для ограничения функциональных возможностей работы с базой (если это необходимо из соображений безопасности). Проектировщики не наполняют базу конкретными данными. Исключение составляет экспериментальное наполнение модельными данными на этапе отладки. Вторая категория исполнителей, работающих с базами данных, — *пользователи*. Они получают исходную базу данных от проектировщиков и занимаются ее наполнением. В общем случае пользователи имеют доступ только к данным, работа с которыми предусмотрена на определенном рабочем месте. Соответственно, СУБД имеет два режима работы: *проектировочный* и *пользовательский*.

Основными типами объектов БД являются таблицы, запросы, формы, отчеты и др.

При проектировании БД методически правильно начинать работу с карандашом и листом бумаги. Сначала разрабатывается техническое задание (ТЗ). Его должен предоставить заказчик. Однако заказчик не всегда грамотен в данной области. Поэтому ему демонстрируют работу аналогичной БД, после чего согласовывают специфику отличий, а если аналога нет, то выясняют круг задач и потребностей заказчика и помогают ему составить ТЗ.

При подготовке технического задания составляют:

- список исходных данных, с которыми работает заказчик;
- список выходных данных, которые необходимы заказчику управления структурой своего предприятия;
- список выходных данных, которые не являются необходимыми для заказчика, но которые он должен предоставлять в другие организации.

При этом очень важно не ограничиваться взаимодействием с головным подразделением заказчика, а провести обсуждение со всеми службами и подразделениями, которые могут оказаться поставщиками данных в базу или их потребителями. Например, при подготовке базы данных для учета абитуриентов в высшем учебном заведении необходимо не только изучить документооборот ректората и всех деканатов, но и понять, что хотели бы получить от базы другие службы. Следует изучить работу подразделений, распределяющих учебную нагрузку преподавателей и отвечающих за распределение аудиторного фонда, за проживание студентов в общежитии и др. В расчет должны приниматься и такие службы, как библиотека, отдел кадров и пр.

Выяснив основную часть данных, которые заказчик потребляет, можно приступать к созданию структуры базы, т. е. структуры ее основных таблиц, по следующему алгоритму:

1. Работа начинается с составления генерального списка полей, он может насчитывать десятки и даже сотни позиций.

2. В соответствии с типом данных, размещаемых в каждом поле, определяют наиболее подходящий тип для каждого поля.

3. Далее распределяют поля генерального списка по базовым таблицам. На первом этапе распределение производят по функциональному признаку. Цель — обеспечить, чтобы ввод данных в одну таблицу производился по возможности в рамках одного подразделения, а еще лучше — на одном рабочем месте. Наметив столько таблиц, сколько подразделений охватывает БД, приступают к дальнейшему делению таблиц. Критерием необходимости деления является факт множественного повтора данных в соседних записях. Например, если в поле «Адрес» наблюдается повтор записей, то таблицу надо поделить на две взаимосвязанные таблицы.

4. В каждой из таблиц намечают ключевое поле — для этого выбирают поле, данные в котором повторяться не могут. Например, для таблицы данных о студентах таким полем может служить индивидуальный шифр студента. Если вообще нет никаких полей, которые можно было бы использовать как ключевые, всегда можно ввести дополнительное поле типа «Счетчик» — оно не может содержать повторяющихся данных.

5. С помощью карандаша и листа бумаги расчерчивают связи между таблицами. На рис. 6.2 показан пример взаимосвязи между группой таблиц, составляющих одну БД. Такой чертеж называется *схемой данных*.

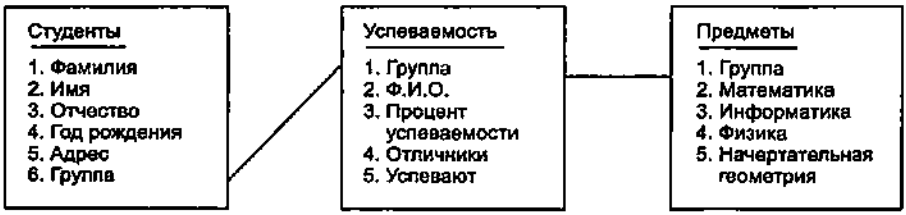


Рис. 6.2. Схема данных

Существует несколько типов возможных связей между таблицами. Наиболее распространенными являются связи «один ко многим» и «один к одному». Связь между таблицами организуется на основе общего поля, причем в одной из таблиц оно обязательно должно быть ключевым. На стороне «один» должно выступать ключевое поле, содержащее уникальные, неповторяющиеся значения. Значения на стороне «многие» могут повторяться.

6. Разработкой схемы данных заканчивается бумажный этап работы над техническим предложением. Схему согласовывают с заказчиком, после чего приступают к непосредственному созданию БД под управлением выбранной системы управления базами данных (СУБД).

Современные СУБД — это системы, которые специализируются на управлении массивами информации одним или множеством одновременно работающих пользователей. Среди наиболее известных СУБД можно отметить:

- иерархические: IMS (Information Management System) фирмы IBM, «Ока» и «ИНЭС» — отечественные;
- сетевые, реляционные: MS Access, Lotus Approach, Borland dBase, Borland Paradox, MS Visual FoxPro, MS SQL Server, Oracle.

Современные реляционные СУБД обеспечивают набор средств для поддержки таблиц и отношений между связанными таблицами, развитый пользовательский интерфейс и средства программирования высокого уровня.

СУБД различаются по своим возможностям и требованиям к вычислительной технике. Различают два основных класса СУБД:

- персональные — ориентированы на работу одного пользователя на ПК (dBase, FoxPro, MS Access и др.);
- многопользовательские — ориентированы на параллельную работу многих пользователей на больших компьютерах (MS SQL Server).

Персональная СУБД имеет удобный интерфейс и применяется как единая программа.

Информация БД размещается в файлах (в реляционных БД — в табличных файлах). Часто эти СУБД приспособлены для работы в сетевой среде, что дает возможность разместить файлы базы данных на файловом сервере и иметь доступ к этой информации всем пользователям, компьютеры которых включены в локальную сеть. Но при этом могут возникнуть большие трудности при одновременной работе нескольких пользователей с одними и теми же данными.

Многопользовательские СУБД состоят из ядра (сервера) и большого числа программ-агентов, которые обслуживают запросы конечных пользователей, и прикладных программ. Ядро и данные находятся на одном компьютере. Одна копия СУБД управляет одной копией данных. Одновременный доступ к данным многих пользователей и устранение конфликтов организует единая управляющая система.

Существующие системы управления базами данных могут по-разному работать с разными объектами и предоставляют пользователю различные функции и средства. Но большинство СУБД опираются на единый комплекс основных понятий. Это дает нам возможность рассмотреть одну систему и обобщить ее понятия, приемы и методы на весь класс СУБД. В качестве такого объекта выберем СУБД Microsoft Access, входящую в пакет Microsoft Office, а в разделе 9.1 приведен пример создания БД под управлением СУБД FoxPro.

СУБД предоставляет несколько средств создания каждого основного объекта БД: ручные (режимы ввода и Конструктора), автоматизированные (режим Мастера), автоматические (средства ускоренной разработки простейших объектов). Целесообразно пользоваться разными средствами.

Поля базы данных не просто определяют структуру базы — они определяют еще и *групповые свойства* данных, записываемых в ячейки, принадлежащие каждому из полей. Например, основные свойства полей таблиц БД СУБД Microsoft Access такие:

- **Имя поля** — определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц);
- **Тип поля** — определяет тип данных, которые могут содержаться в данном поле;

- **Размер поля** — определяет предельную длину (в символах) данных, которые могут размещаться в данном поле;
- **Формат поля** — определяет способ форматирования данных в ячейках, принадлежащих полю;
- **Маска ввода** — определяет форму, в которой вводятся данные в поле (средство автоматизации ввода данных);
- **Подпись** — определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство **Имя поля**);
- **Значение по умолчанию** — то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных);
- **Условие на значение** — ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты);
- **Сообщение об ошибке** — текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных (проверка ошибочности выполняется автоматически, если задано свойство **Условие на значение**);
- **Обязательное поле** — свойство, определяющее обязательность заполнения данного поля при наполнении базы;
- **Пустые строки** — свойство, разрешающее ввод пустых строковых данных (от свойства **Обязательное поле** отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например к текстовым);
- **Индексированное поле** — если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значения в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Свойства у полей могут различаться в зависимости от типа данных. Например, список указанных свойств полей относится в основном к полям текстового типа. Поля других типов могут иметь или не иметь эти свойства, но могут добавлять к ним и свои. Например, для данных, представляющих действительные числа, свойством является количество знаков после десятичной запятой. Для полей, используемых для хранения рисунков, зву-

козаписей, видеоклипов и других OLE-объектов, большинство вышеуказанных свойств не имеет смысла.

Таблицы баз данных, как правило, допускают работу с большим количеством разных *типов данных*. Например, БД Access работает со следующими типами данных: текстовый, Мемо, числовой, дата/время, денежный, счетчик, логический, OLE, гиперссылка, Мастер подстановок. О таблицах БД Access подробнее см. раздел 6.1.

Поле МЕМО — используется для хранения указателя длиной до 65 535 символов (данные хранятся в другом месте).

«Счетчик» — числовое поле, имеющее свойство автоматического перемещения на 1.

Логический — тип для хранения логических данных (могут принимать только два значения, например «Да» или «Нет»).

Поле объекта OLE — специальный тип данных, предназначенный для хранения OLE-объектов, например мультимедийных. Реально, конечно, такие объекты в таблице не хранятся. Как и в случае полей МЕМО, они хранятся в другом месте внутренней структуры файла базы данных, а в таблице хранятся только указатели на них (иначе работа с таблицами была бы чрезвычайно замедленной).

Гиперссылка — специальное поле для хранения URL-адресов для веб-объектов Интернета. При щелчке по ссылке автоматически происходит запуск браузера и воспроизведение объекта в его окне.

Мастер подстановок — это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбирать из раскрывающегося списка.

Запросы — удобное средство доступа к данным. О запросах подробнее см. раздел 6.2.

Запросы могут быть на выборку, с параметром, фильтр, итоговые, вычисления в запросах, запросы на изменение. Все виды запросов подробно описаны в методических указаниях к соответствующим лабораторным работам в отдельной книге [8].

Программа MS Access также позволяет создавать *отчеты*. Об отчетах см. подробнее раздел 6.3.

Для баз данных предъявляются особые требования с точки зрения их *безопасности*, поэтому в них реализован особый подход к сохранению данных.

Базы данных — это особые структуры. Информация, которая в них хранится, очень часто имеет общественную ценность. Поэтому целостность содержимого базы не может и не должна зависеть ни от определенных действий некоего пользователя, забывшего сохранить файл перед выключением компьютера, ни от перебоев в электросети.

Проблема безопасности баз данных решается тем, что в СУБД для сохранения информации используется двойной подход. В части операций, как обычно, участвует ОС, но некоторые операции сохранения происходят в обход операционной системы. Операции изменения структуры БД, создания новых таблиц или новых объектов происходят при сохранении файла базы данных. Об этих операциях СУБД предупреждает пользователя. Это, так сказать, глобальные операции. Их никогда не проводят с БД, находящейся в коммерческой эксплуатации, только с ее копией. В этом случае любые сбои в работе вычислительных систем не страшны.

Контрольные вопросы

1. Что такое данные, информация и знания?
2. Что такое база данных и банк данных, база знаний?
3. По каким моделям строят БД?
4. Какими способами можно создать компьютерную модель БД?
5. Что такое нормализация таблиц БД?
6. Что такое схема данных?
7. Что такое СУБД? Приведите примеры.
8. Каковы основные свойства полей СУБД MS Access?
9. С какими типами данных работает MS Access?
10. Каковы основные объекты БД MS Access?
11. Из чего состоит таблица БД и в каких режимах ее можно создать?
12. Как обеспечивается безопасность баз данных?

6.1. Создание БД под управлением СУБД MS Access

СУБД MS Access служит инструментом для работы с базой данных, состоящей из набора таблиц, форм, запросов и отчетов, которые используются для представления и обработки данных.

Работа с этими объектами БД выполняется в *окне базы данных.*

Программа MS Access запускается активацией пиктограммы с изображением ключа на панели Microsoft Office.

Таблица — основной структурный элемент реляционной СУБД. Она состоит из полей (столбцов) определенного типа и записей (строк). В нижней части окна имеется строка, называемая *полем номера записи*, которая содержит *кнопки перехода*, с помощью которых осуществляется перемещение по таблице. Каждая запись имеет слева кнопку — *маркер записи*, с помощью которого запись можно выделить. Операции с записями можно осуществлять с помощью команд контекстного меню. Все изменения в записях сохраняются автоматически в режиме реального времени.

Маркер таблицы находится в ее верхнем левом углу. Выделить всю таблицу можно щелчком левой кнопки мыши по маркеру, а щелчок правой кнопкой откроет контекстное меню. Переместить столбец можно, выделив его (щелчок левой кнопкой мыши по заголовку) и перетащив заголовок с помощью мыши на новое место. Убрать столбец можно командами меню **Формат → Скрыть столбцы**. Восстановить столбец можно двойным щелчком по границе между столбцами, где он был скрыт. Размеры столбцов и строк можно менять с помощью мыши.

Создание БД *в режиме Мастера*: при запуске Access открывается диалоговое окно. Следует выбрать вариант **Запуск мастера**. Дважды щелкнуть мышью по ярлыку шаблона базы данных. Указать имя и каталог создаваемой БД.

В режиме таблицы открывается заготовка, в которую сразу можно вводить информацию. При сохранении такой таблицы Access автоматически присвоит типы данных и формат каждому полю.

В режиме Конструктора можно самостоятельно задать имена полей, выбрать их тип и настроить свойства. Перейдя с помощью кнопки **Вид** в режим конструктора, можно вставить новое поле, наведя указатель мыши на маркер поля и нажав клавишу **Insert**, удалить — **Delete**. Закончить создание структуры таблицы следует щелчком по кнопке **Вид** и переходом в режим таблицы для заполнения ее данными.

Схема данных — специальное окно для создания связей между таблицами, открывается с помощью команд **Сервис → Схема данных** или щелчком по одноименной кнопке на панели инстру-

ментов. В открывшемся диалоговом окне **Добавление таблицы** следует выбрать таблицы для включения их в структуру межтабличных связей. Если связи уже задавались, то, открыв контекстное меню щелчком правой кнопки мыши, следует выбрать пункт **Добавить таблицу**.

Связи между полями таблицы создаются перетаскиванием имени поля из одной таблицы в другую на соответствующее ему связываемое поле. В диалоговом окне **Связи** можно задать свойства образующейся связи. Для защиты от случайного удаления записей в одной из таблиц следует включить флажок **Обеспечение условия целостности данных**.

Для обеспечения целостности ключевым выбирают поле основной таблицы, связываемые поля должны иметь одинаковый тип.

Для одновременного изменения или удаления данных во всех связанных таблицах при изменении их в главной таблице следует включить флажки **Каскадное обновление связанных полей** и **Каскадное удаление связанных записей**.

Форма — это средство, используемое для ввода и представления данных таблицы на экране компьютера. Она представляет собой электронный бланк с полями для ввода данных, которые затем автоматически заносятся в таблицы базы. Использование форм дает более высокую степень контроля над представлением данных, чем в таблице, наличие разнообразных элементов управления данными и их свойствами.

Виды форм:

- *в столбец*, когда поля записи расположены в столбец. Форма отображает одновременно одну запись. Для перехода к другой записи используют панель **Запись** в нижней части окна формы. Номер искомой записи вводится в поле номера записи, а перемещение по записям осуществляется с помощью стрелок;
- *ленточная*, когда одновременно отображается несколько записей;
- *табличная*.

Структура формы имеет три раздела: заголовок, область данных, примечания.

В области данных содержатся *элементы управления*:

- *связанное поле* (вводимые в него данные поступают и в одноименное поле таблицы, на базе которой создана форма). Для создания связанных полей служит элемент **Поле** на панели элементов;

- *присоединенная надпись* (перемещается вместе со своим элементом);
- *переключатели*, с которыми можно связать разные команды;
- *флажки* — действуют как переключатели, но допускают множественный выбор, удобны для сортировки данных;
- *список* — содержит фиксированный набор значений, позволяет не вводить данные, а выбирать их из списка;
- *поле со списком* — применяется так же, как и список, но занимает меньше места в форме;
- *командные кнопки*, с которыми можно связать полезные команды, например поиска записи, перехода между записями и др.;
- *вкладки* — позволяют разместить много информации на ограниченной площади;
- *поле объекта OLE* — служит для размещения внешнего объекта (иллюстрации, видеозаписи и др., созданных разными приложениями). С присоединенной рамкой связано одно из полей таблицы, содержимое которого в ней отображается и может меняться от записи к записи.

Формы отображаются в режимах формы, конструктора, таблицы, предварительного просмотра.

Для создания новых и изменения существующих форм можно использовать *режим Конструктора* (создание формы с нуля, но сложный путь). Из этого режима можно переходить в другие режимы, улучшать внешний вид и эффективность работы с формой. Созданные элементы управления формы следует выровнять с помощью команд **Форма → Выровнять**.

С помощью Мастера по созданию форм (более простой путь) можно самостоятельно разработать форму или создать ее в 4 этапа: выбор полей, данные для которых можно будет вводить в форме; выбор внешнего вида формы; выбор фонового рисунка формы; задание имени формы. Мастер автоматически выполнит большинство операций, задавая вопросы, на которые следует ответить. Создав макет формы с помощью Мастера, можно перейти в режим Конструктора для продолжения самостоятельной разработки формы.

Для создания простой формы с одним столбцом (полем) следует использовать кнопку **Новый объект**. Для изменения всей формы следует нажать на кнопку **Автоформат** на панели инструментов и выбрать новый внешний вид формы.

Для изменения поля базовой таблицы, с которой связан элемент управления, следует навести указатель мыши на элемент

управления и выполнить двойной щелчок. В появившемся окне свойств в поле **Данные** из списка нужно выбрать другое имя поля базовой таблицы.

Чтобы добавить рисунок в форму, в окне свойств формы следует выбрать свойство **Рисунок** и открыть построитель рисунков или указать имя файла нового рисунка.

Изменить внешний вид элементов управления (например, поля) можно, выделив его (инструмент **Выбор объектов**) и задав новые параметры (например, размер или вид шрифта) с помощью элементов панели форматирования. Изменить текст подписи можно, выделив старый текст и введя новый. Щелчок правой кнопкой мыши открывает контекстное меню, в котором имеются дополнительные возможности изменения оформления. Создать надпись можно, перетащив вниз разделительную границу между заголовком и областью данных. Вверху освобождается достаточно места для создания крупной надписи. Щелкнув на панели элементов по элементу **Надпись**, а затем по форме, получают текстовую рамку для ввода текста. Оформить текст можно после его ввода.

Копирование оформления одного элемента управления для другого можно выполнить, выделив первый элемент, нажав кнопку **Копировать формат**, а затем выбрав элемент управления, у которого нужно изменить оформление.

Для овладения практическими навыками создадим простейшую БД «Студенты» под управлением СУБД Access, включая:

- построение таблиц базы данных;
- создание и редактирование форм в разных режимах;
- связывание таблиц в БД с помощью схемы данных.

Последовательность действий:

1. Запускаем программу MS Access. Создаем новую базу данных под именем «Студенты» и сохраняем ее.

2. Изучаем основные виды объектов в диалоговом окне создания базы данных и режимы работы (рис. 6.3).

3. Создаем в режимах Конструктора и Мастера таблиц три таблицы: «Успеваемость», «Адрес_Специализация», «Список рассылки».

4. Вводим в таблицы данные (рис. 6.4).

5. С помощью схемы данных устанавливаем связи между таблицами по ключевым полям (рис. 6.5).

6. Создаем формы для удобного ввода данных (рис. 6.6).

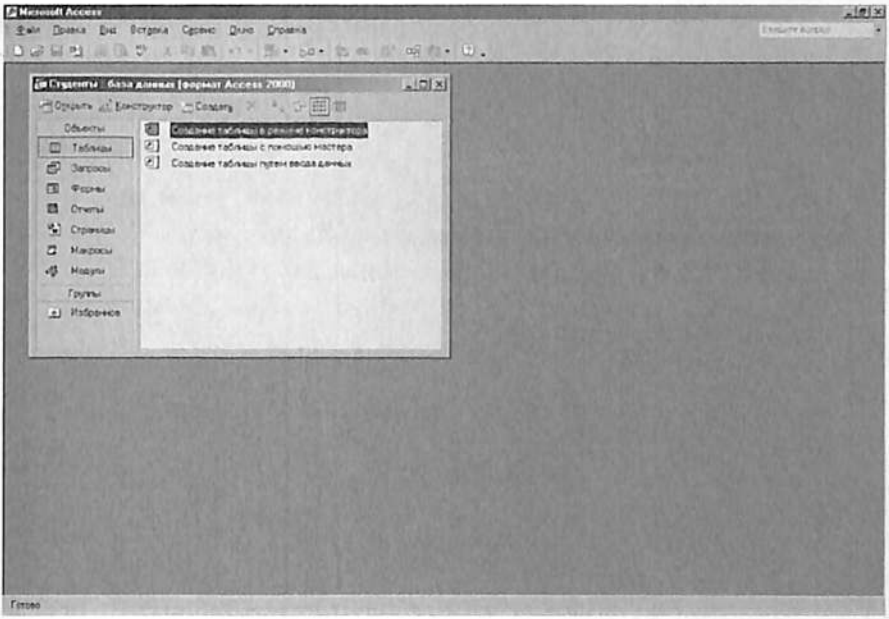


Рис. 6.3. Диалоговое окно создания новой БД в программе Access

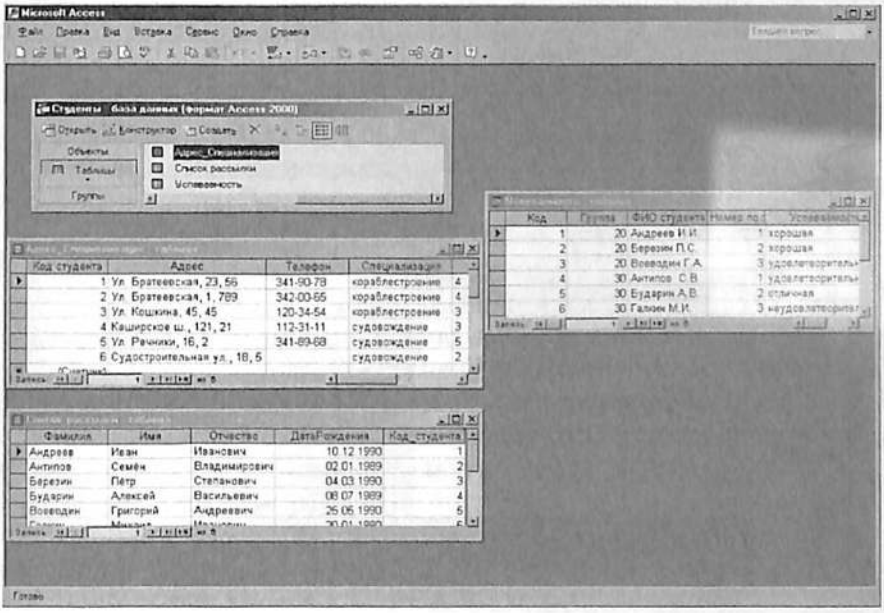


Рис. 6.4. Ввод данных в таблицы

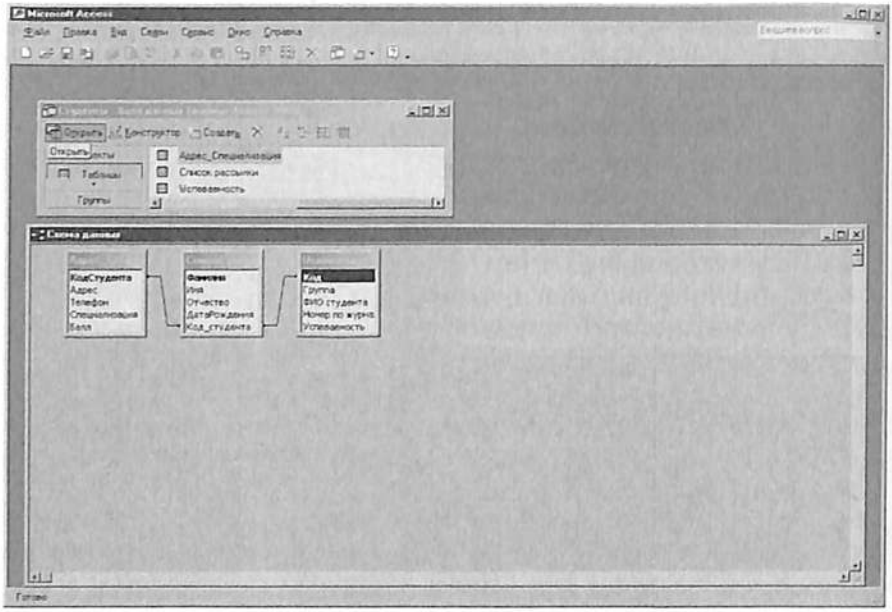


Рис. 6.5. Связывание таблиц по схеме данных

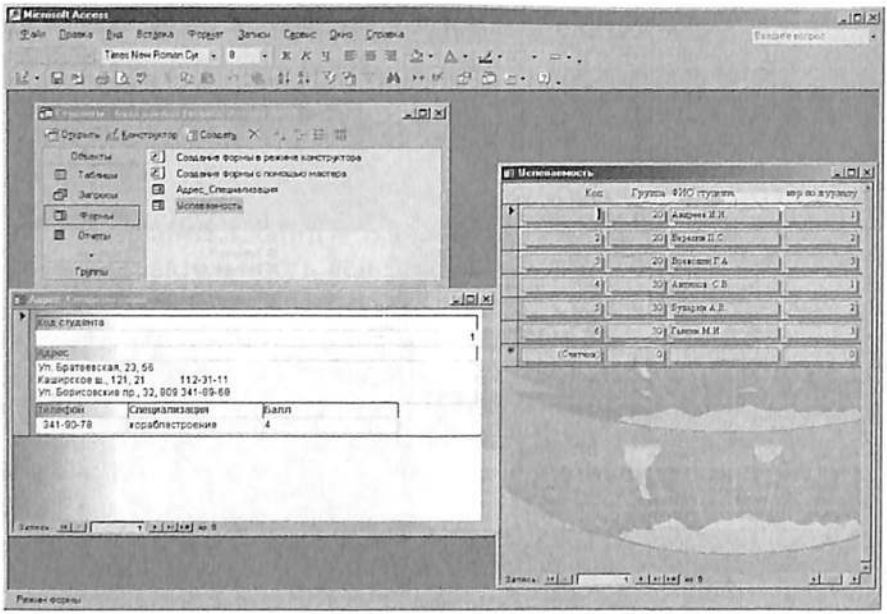


Рис. 6.6. Создание форм

Контрольные вопросы

1. Каковы назначение и основные функции СУБД Access?
2. Каковы режимы создания БД?
3. Как связать таблицы в БД?
4. Каким должно быть поле первичного ключа?
5. Какие вы знаете режимы создания таблиц в MS Access?
6. Какие режимы создания и редактирования форм вы знаете?
7. Как добавить рисунок в форму?

6.2. Создание и использование запросов в БД

Для одной и той же таблицы можно создать множество запросов и с их помощью извлечь запрашиваемую информацию из БД. Запросы могут быть простыми и сложными. В сложных запросах выполняется последовательное сравнение содержимого полей и учитываются определенные условия. Данные при запросе можно сортировать, фильтровать (отсеивать), объединять, разделять, изменять, выполнять вычисления.

Результирующая таблица содержит информацию, выдаваемую из БД по запросу. *Бланк запроса* предназначен для формирования запроса — ввода используемых полей и условий или с помощью перетаскивания между окнами элементов запроса. Запросы предназначены для создания *результирующей таблицы*, в которой отображаются данные из базовых таблиц по условию запроса.

Запросы создаются вручную или автоматически с помощью Мастера. Для их создания следует открыть вкладку **Запросы** диалогового окна **База данных** и щелкнуть по кнопке **Создать**. В диалоговом окне **Новый запрос** выбрать **Конструктор**, т. е. ручной режим. В диалоговом окне **Добавление таблиц** выбрать те таблицы, на которых будет основан запрос, и занести их в верхнюю половину поля запроса, щелкнув по кнопке **Добавить**. Другие вкладки в этом окне используют, если запрос основывается на ранее созданном запросе.

Бланк запроса по образцу имеет две панели: на верхней — списки полей тех таблиц, на которых основан запрос; на нижней — структура запроса, строки результирующей таблицы: **Поле**, **Имя Таблицы**, **Сортировка**, **Вывод на экран**, **Условие отбора**.

Запуск запроса (вывод результирующей таблицы) осуществляется щелчком по кнопке **Вид**. Повторный щелчок — выход из результирующей таблицы на новый запрос.

Запросы с параметром — выбор параметра по желанию пользователя. Для этих запросов используют команду языка SQL (Structured Query Language — структурированный язык запросов) `LIKE [любой текст]`, например: `LIKE [Введите имя автора]`. Команду `LIKE` набирают в строке **Условие отбора** в поле выбора. После запуска запроса в открывшемся диалоговом окне необходимо ввести параметр, например «Иванов». Выдается результирующая таблица, содержащая информацию по запросу, относящуюся к автору Иванову. Если была запрошена информация о произведениях Иванова, выпущенных в 1995 г., то она будет выбрана из таблицы, где содержатся все сведения о его произведениях за многие годы.

Фильтр — набор условий, применяемых для отбора подмножества записей или их сортировки. Фильтры могут быть простыми, содержащими условие отбора для одного поля, и сложными, содержащими несколько условий для различных полей. *Условие отбора* — это ограничение, накладываемое на запрос или фильтр для отбора конкретных записей. Условия отбора создаются с использованием операторов сравнения (=, >, < и др.). В сложных фильтрах условия отбора между собой связываются операцией логического умножения И. Оператор `AND` применяют при объединении условий в одной строке, а `OR` — когда выражения находятся в разных строках бланка. Для того чтобы задать в бланке условие отбора для конкретного поля, нужно ввести выражение (любую комбинацию операторов, констант или их имен, функций, имен полей и т. д., результатом которой является конкретное значение) в ячейку строки **Условие отбора** для этого поля.

Имеется три типа фильтров: фильтр по выделенному фрагменту, обычный и расширенный. *Фильтр по выделенному фрагменту* — средство создания фильтра, в котором выделенные записи (или их части) используются для отбора. Для создания *обычного* фильтра в окне таблицы следует выбрать команды **Записи → Фильтр → Изменить фильтр**. В появившемся окне ввести условия поиска в соответствующих полях. Далее выполнить команду **Применить фильтр**. *Расширенный фильтр* — простейший бланк запроса для вывода некоторого подмножества записей таблицы. Для его создания в окне таблицы следует выполнить

команды **Записи** → **Фильтр** → **Расширенный фильтр**. В появившемся окне следует ввести в ячейки строки **Условия отбора** фильтр, который позволит ограничить записи вывода. Здесь же можно задать условия сортировки. Для получения результата следует нажать кнопку **Применить**.

Для получения практических навыков работы по созданию запросов в базе данных под управлением СУБД Access используем БД «Студенты», созданную в разделе 6.1.1:

1. Запускаем программу MS Access. Открываем базу данных под именем «Студенты» и в диалоговом окне **База данных** выбираем вкладку **Запросы**.

2. Составляем запросы на выборку, заполняя бланк запроса по образцу.

3. *Запрос на выборку.* В запросе участвуют все три таблицы. Запрос состоит из 4 полей: «ФИО студента», «Дата рождения», «Успеваемость», «Телефон». После выбора таблиц установлено условие на вывод значений: успеваемость хорошая (рис. 6.7).

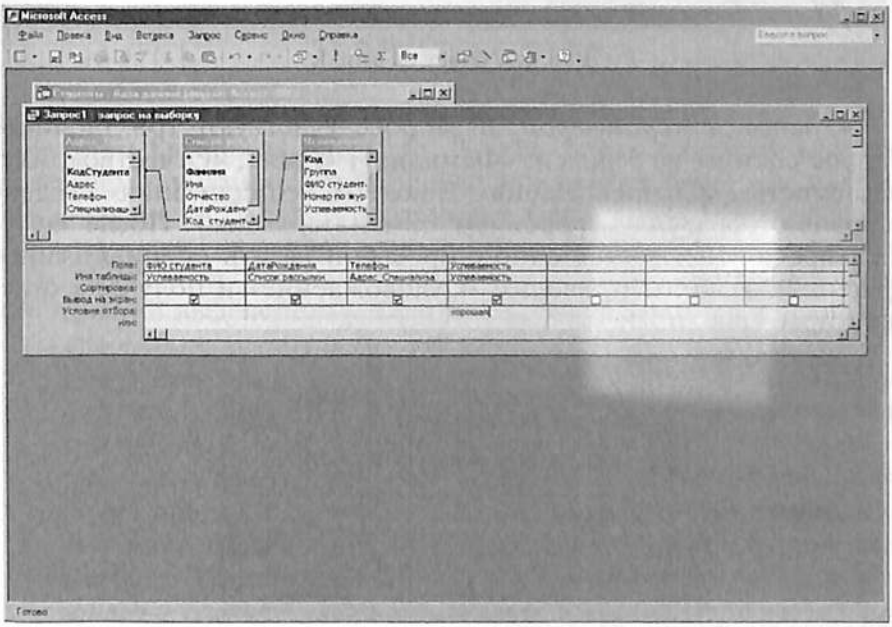


Рис. 6.7. Бланк запроса

После выполнения запроса получена таблица с ответом (рис. 6.8).

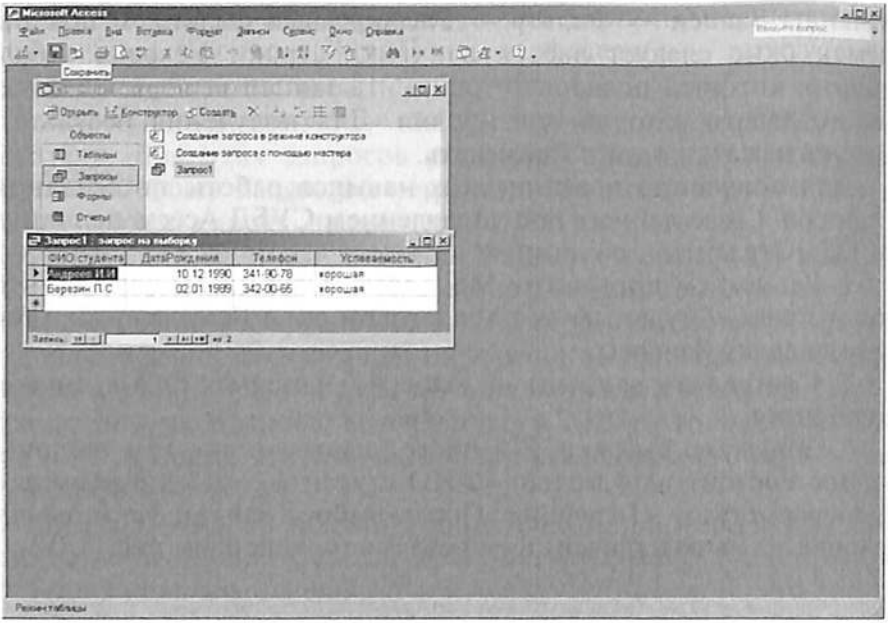


Рис. 6.8. Ответ на запрос

4. *Запрос с параметром.* В запросе участвуют три таблицы. Запрос состоит из 5 полей: «Фамилия», «Имя», «Отчество», «Успеваемость», «Специализация». В поле «Специализация» введено условие отбора: `LIKE [Введите специализацию]`. После запуска запроса открывается окно с просьбой ввести специализацию (рис. 6.9). Например, вводим «Судовождение» и получаем ответ на запрос (рис. 6.10).

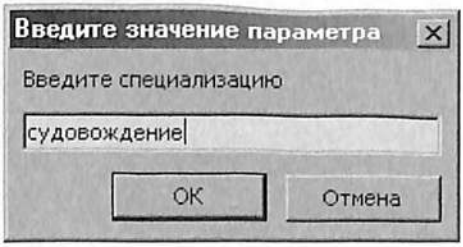


Рис. 6.9. Окно ввода специализации

5. *Запрос итоговый.* В запросе участвует одна таблица. Для создания итогового запроса нажимаем кнопку суммы на панели

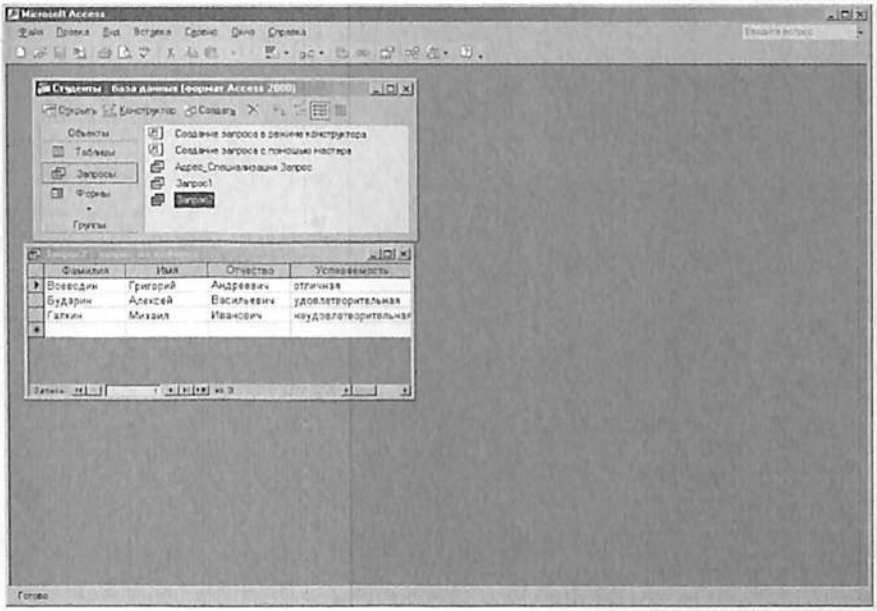


Рис. 6.10. Ответ на запрос

Конструктор запросов. Появляется дополнительная строка под названием **Групповая операция**. В этой строке в поле **Балл** устанавливаем функцию **Avg** (Среднее арифметическое) для нахождения среднего балла успеваемости (рис. 6.11). После запуска получаем ответ на запрос (рис. 6.12).

6. Используя простые и сложные условия отбора, создаем фильтры: по выделенному фрагменту, обычный и расширенный:

- *фильтр по выделенному значению*. Открываем таблицу «Адрес_Специализация», выделяем название улицы «Судостроительная» и щелкаем мышью по кнопке **Фильтр по выделенному**. Результат представлен на рис. 6.13;
- *фильтр обычный*. Открываем таблицу «Список рассылки». Выполняем команды **Записи** → **Фильтр** → **Изменить фильтр**. В строке «Успеваемость» пишем — «отличная». Выполняем команду **Применить фильтр**. Получаем результат (рис. 6.14);
- *фильтр расширенный*. Открываем таблицу «Успеваемость». Выполняем команды **Записи** → **Фильтр** → **Расширенный фильтр**. Установим поле «Успеваемость» и поставим условие — «неудовлетворительная», в поле «Группа» установим 30. Бланк запроса и результат по запросу см. на рис. 6.15.

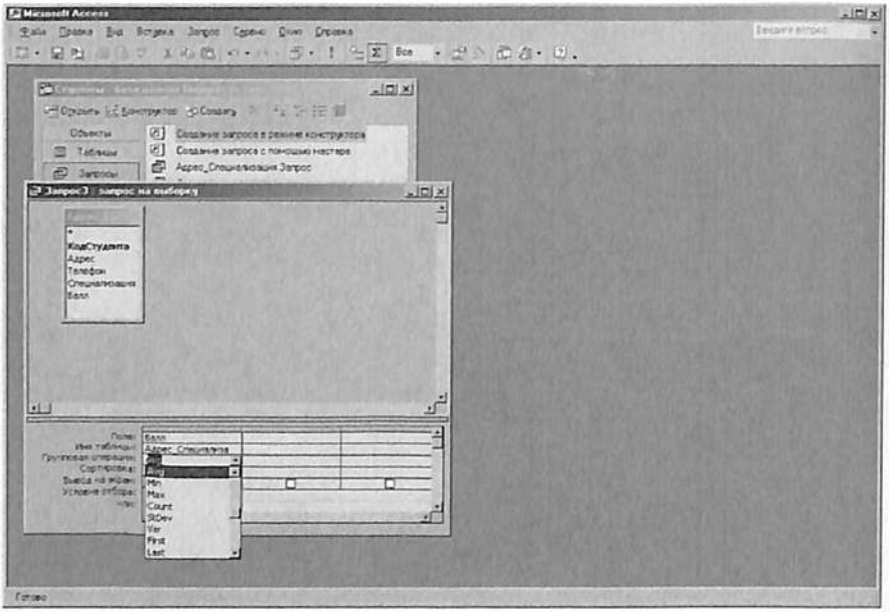


Рис. 6.11. Бланк итогового запроса

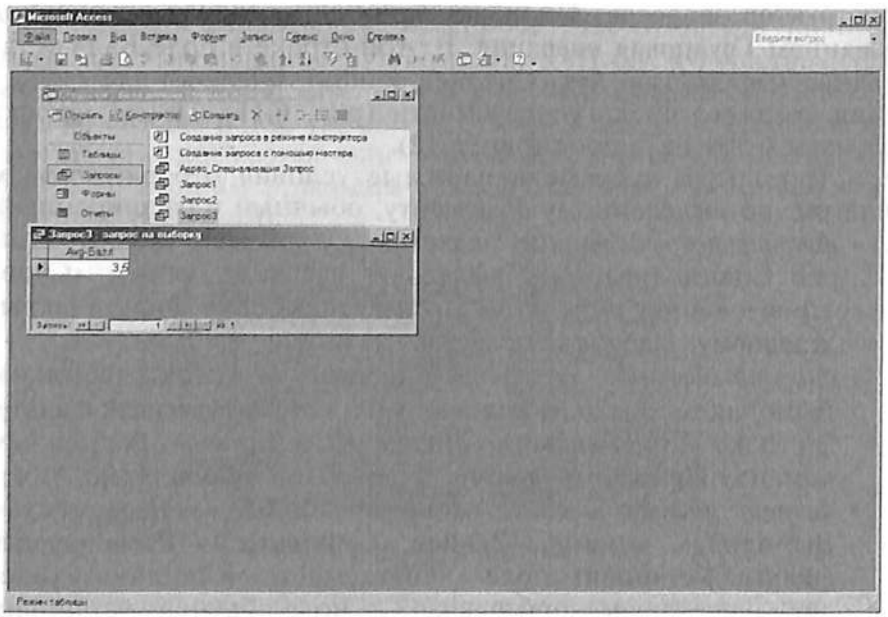


Рис. 6.12. Ответ на итоговый запрос

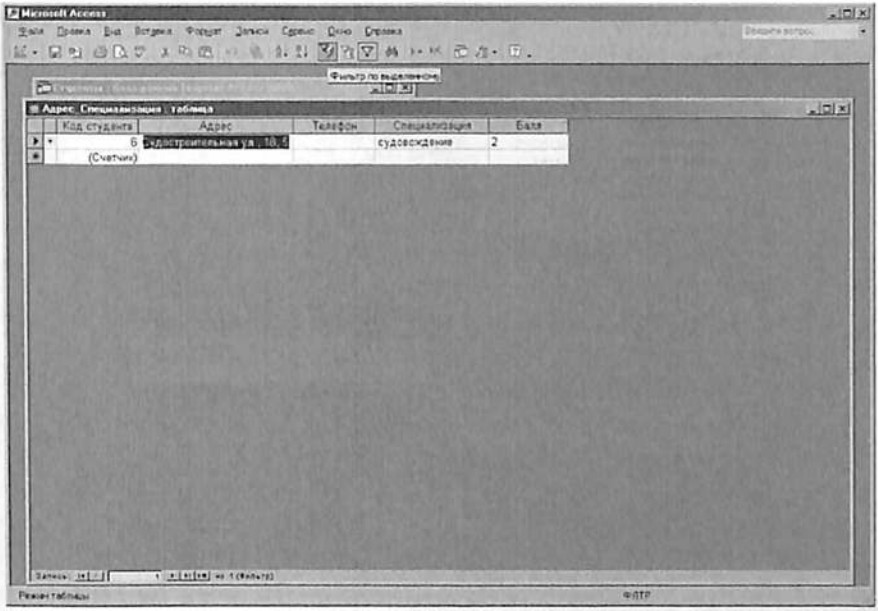


Рис. 6.13. Фильтр по выделенному

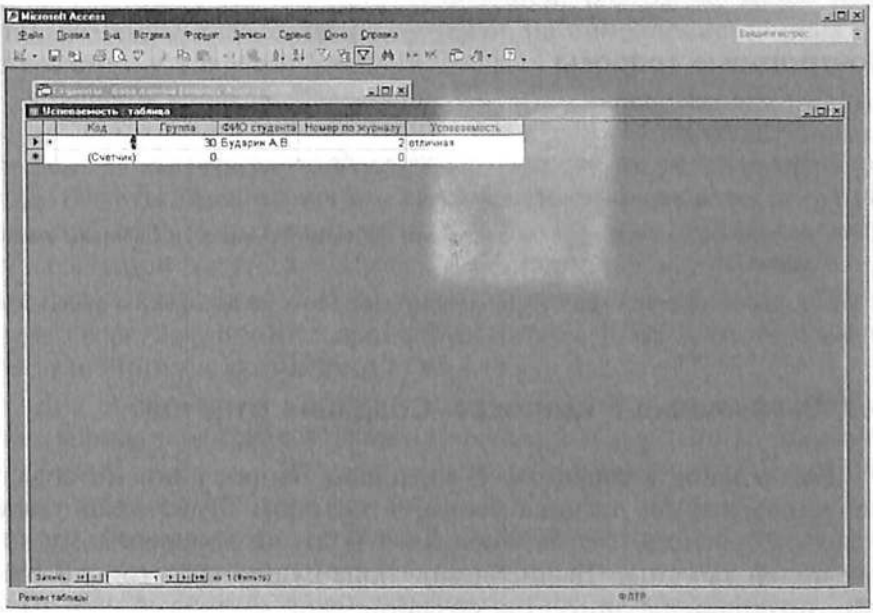


Рис. 6.14. Обычный фильтр

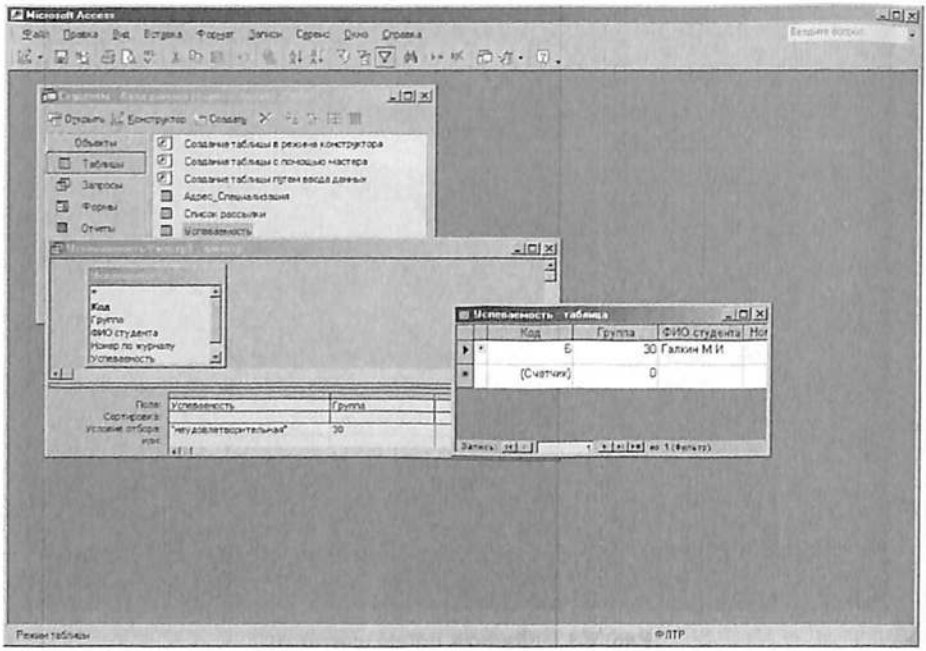


Рис. 6.15. Бланк запроса и результат запроса

Контрольные вопросы

1. Какие виды запросов вы знаете?
2. Что такое результирующая таблица и бланк запроса?
3. Что такое условия отбора данных?
4. Что представляют собой фильтры: по выделенному, обычный, расширенный?
5. В чем сходство и различие между запросом на выборку и фильтром?

6.3. Вычисления в запросах. Создание отчетов

Вычисления в запросах. С помощью запроса можно создать поле, называемое вычисляемым, в котором будет представлен результат расчета (сумма, среднее и т. п.) по содержимому других полей таблицы. Вычисляемое поле существует только в результирующей таблице. Для создания запроса используют *бланк запроса по образцу*. Разница только в том, что в одном из столб-

цов вместо имени поля следует записать формулу, в которую входят названия полей (в квадратных скобках), содержащих исходные данные и знаки математических операций. Например,

Цена: [Истрачено (сумма)]/[Количество изделий]
название: [Поле 1] Знак [Поле 2]

Для записи длинной формулы, нажав сочетание клавиш **Shift+F2**, следует открыть диалоговое окно **Область ввода**. После ввода формулы щелчком по кнопке **OK** ее можно перенести в бланк запроса по образцу. Результаты расчетов будут представлены в результирующей таблице при включении отображения вычисляемого поля.

Запросы на изменение позволяют автоматически создавать новые или изменять имеющиеся таблицы. Запрос на создание таблицы составляют с помощью запроса на выборку, в который войдут все поля базовой таблицы плюс новое вычисляемое поле. Щелкнув по кнопке **Вид**, следует убедиться, что создана *результирующая таблица*, более полная, чем базовая. Далее в меню **Запрос** в режиме Конструктора выбирают команду на создание базовой таблицы, равной результирующей. В этом же меню имеются команды для создания запросов на обновление данных, на добавление и удаление записей.

Создание отчетов. *Отчет* — это гибкое средство для организации данных при выводе на печать в требуемом виде. Отчеты создаются на основе таблиц, форм, запросов или инструкции SQL. Отчеты предназначены для вывода информации на принтер, поэтому принтер должен быть установлен. Отчет можно создать и при отсутствии принтера. Следует выполнить команды **Пуск → Настройка → Принтеры → Установка принтера**. Далее нужно зарегистрировать драйвер принтера, взяв один из драйверов, входящих в дистрибутив ОС.

Для составления отчетов следует выбрать в диалоговом окне **База данных** вкладку **Отчеты** и щелкнуть по кнопке **Создать**. В диалоговом окне **Новый отчет** можно создать автоотчет, используя режим Мастера или вручную. Мастер автоматически выполнит большинство операций, задавая вопросы, на которые пользователь должен ответить. Со структурой отчета можно ознакомиться, создав какой-либо автоотчет, а затем открыв его в режиме Конструктора.

Структура отчета состоит из 5 разделов: заголовка, верхнего колонтитула, области данных, нижнего колонтитула, примечания.

Раздел заголовка — дается общий заголовок отчета. Раздел верхнего колонтитула используется для печати подзаголовков, номеров страниц (колонцифры), логотипов и др. Область данных предназначена для размещения элементов управления, в которые выдаются данные из таблиц базы для печати на принтере. Раздел нижнего колонтитула используется для печати таких же данных, как и в верхнем колонтитуле. При этом могут использоваться встроенные функции для вывода текущей даты и страниц (функция `Now()`, `Page()`, `Pages()`) и др. Оператор & (оператор конкатенации) служит при этом для «склеивания» текста, заключенного в кавычки, со значениями, возвращаемыми функциями. Раздел примечаний предназначен для дополнительной информации.

Для получения практических навыков работы по вычислениям в запросах, включая создание вычисляемого поля и итоговые вычисления, и созданию отчетов на экране и в виде бумажного документа в базе данных под управлением СУБД Access используем БД «Студенты», созданную в разделе 6.1:

1. Запускаем программу MS Access. Открываем базу данных «Студенты».

2. Выполняем вычисления в запросах. С помощью запроса создаем вычисляемое поле «Успев_%» (набор баллов в процентах каждым студентом), в котором представлен результат расчета по содержимому поля «Балл» таблицы «Адрес_Специализация». Для создания запроса заполняем *бланк запроса по образцу*. В вычисляемом поле вместо имени поля записываем формулу — Успев_%: [Балл]/21*100. Результаты расчетов представлены в результирующей таблице (рис. 6.16).

3. На основе таблиц создаем отчеты. В окне базы данных выбираем объект **Отчеты** и нажимаем кнопку **Создать**. В появившемся окне выбираем вариант **Автоотчет ленточный** для таблицы «Адрес_Специализация» (рис. 6.17).

4. Создаем отчет с помощью Мастера отчетов. Выбираем таблицу «Успеваемость» и в ней 4 поля, которые будут входить в отчет. Нажимаем кнопку **Далее**. Затем в диалоговых окнах определяем уровни группировки, порядок полей и сортировку данных в полях. Выбираем табличный макет, альбомную ориентацию

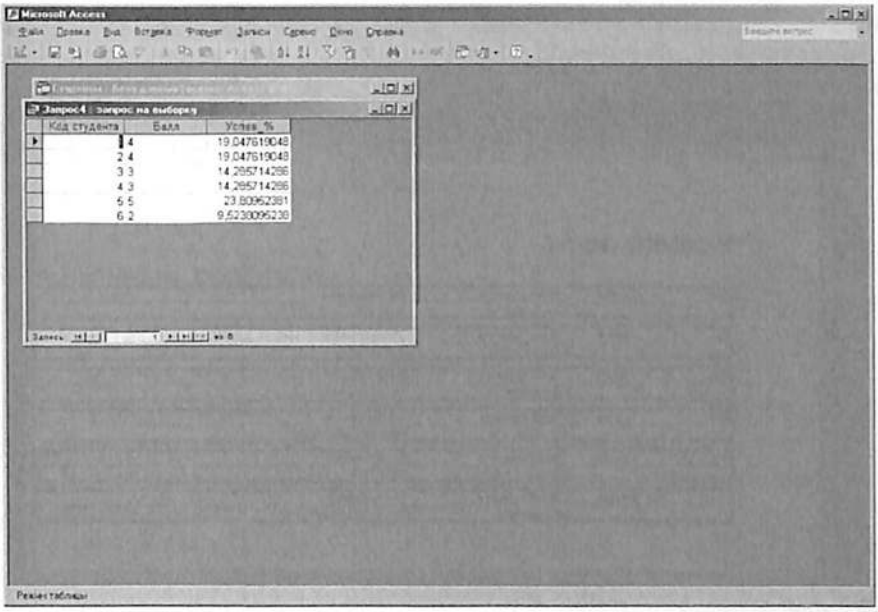


Рис. 6.16. Вычисления в запросах

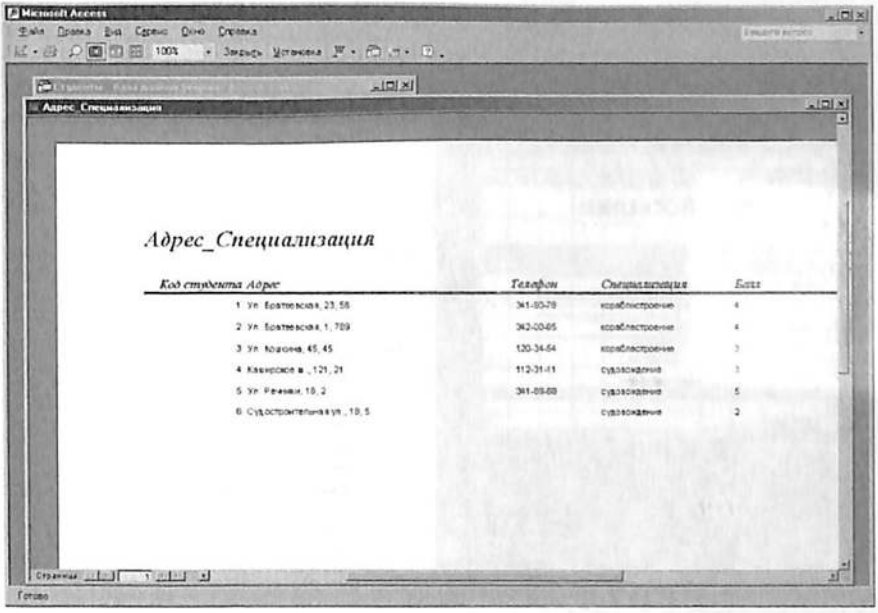


Рис. 6.17. Автоотчет ленточный

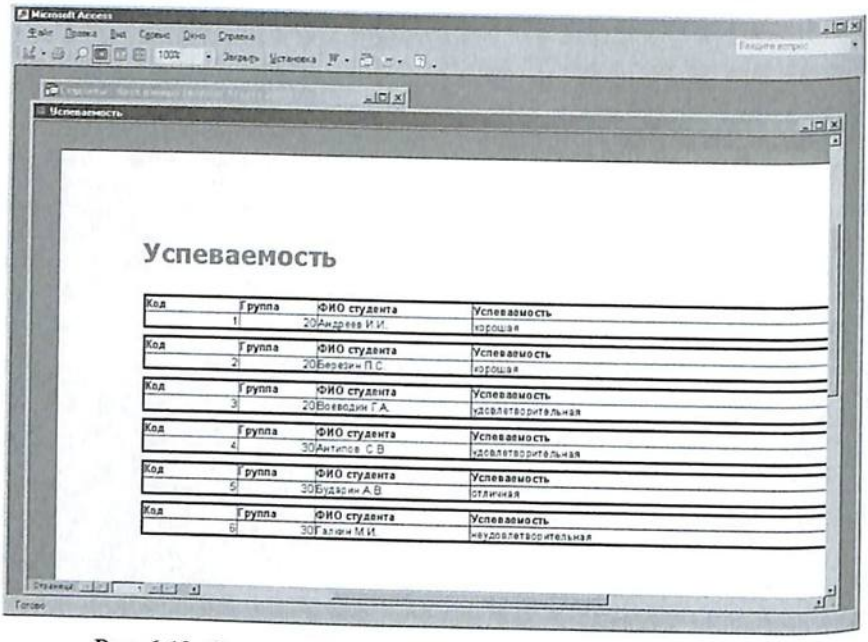


Рис. 6.18. Отчет, составленный с помощью Мастера отчетов

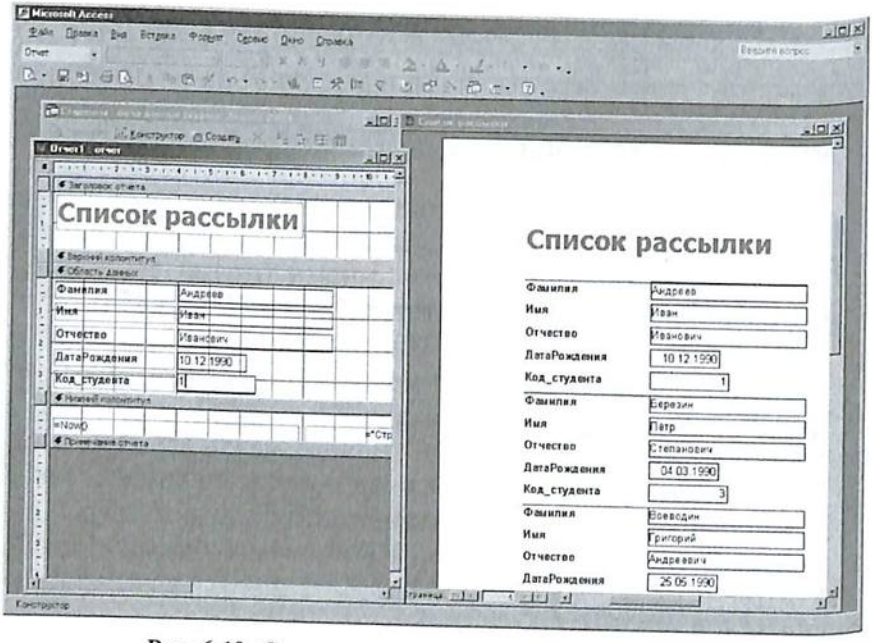


Рис. 6.19. Создание отчета в режиме Конструктора

стиль заголовка Обычный. Нажимаем кнопку **Готово**. Результат представлен на рис. 6.18.

Создаем отчет с помощью Конструктора. Выбираем таблицу «Список рассылки». Создаем автоотчет, а затем открываем его в режиме Конструктора. Заполняем необходимые разделы и отправляем на печать (рис. 6.19).

Контрольные вопросы

1. Что такое вычисляемое поле?
2. Что такое итоговые вычисления?
3. Что такое запрос на изменение?
4. Какие способы генерации отчетов вы знаете?
5. Что общего и в чем различие между разделами отчетов и разделами форм?

Глава 7
СЕТИ ЭВМ, ВЕБ-ТЕХНОЛОГИИ

7.1. Локальные и глобальные сети ЭВМ

Для связи пользователей компьютеров, работающих далеко друг от друга, создаются компьютерные сети, которые позволяют быстро обмениваться информацией. Компьютерная сеть образуется при соединении нескольких компьютеров. *Прямое соединение* двух компьютеров в ОС Windows XP можно обеспечить с помощью стандартного средства ОС командами **Пуск → Программы → Стандартные → Связь → Мастер новых подключений → Установить прямое подключение к другому компьютеру**.

Назначение компьютерных сетей — обеспечение совместного доступа к общим ресурсам. *Ресурсы* бывают трех типов: аппаратные, программные, информационные. При работе в компьютерной сети любого типа одновременно происходит совместное использование всех трех типов ресурсов.

При создании компьютерных сетей важно обеспечить совместимость оборудования (по электрическим и механическим характеристикам), информационного и программного обеспечения (программ и данных, систем кодирования, форматов данных). Стандартизация этих задач основана на модели OSI (Model of Open System Interconnections — модель взаимодействия открытых систем), созданной по предложениям Международного института стандартов ISO (International Standards Organization).

По модели ISO/OSI можно выделить семь уровней архитектуры сетей:

1. *Прикладной* (самый верхний) *уровень*. Пользователь взаимодействует с вычислительной системой, создает документ.

2. *Уровень представления.* ОС компьютера пользователя фиксирует нахождение созданных данных.

3. *Сеансовый уровень.* Компьютер пользователя взаимодействует с сетью. Протоколы проверяют права пользователя на связь и передают документ к протоколам следующего уровня.

4. *Транспортный уровень.* Документ преобразуется в форму передачи данных по сети.

5. *Сетевой уровень* задает маршрут движения данных по сети.

6. *Уровень соединения* модулирует сигналы физического уровня. Эти функции выполняет сетевая карта или модем.

7. *Физический* (самый нижний) *уровень.* Обеспечивает обмен сигналами между устройствами. Реальная передача данных по битам. На компьютере получателя данных процесс преобразования обратный — от битовых сигналов до документа.

Обмен данных происходит путем перемещения их с верхнего уровня на нижний и обратным воспроизведением.

Совместимость на каждом из семи возможных уровней архитектуры компьютерной сети обеспечивается действующими специальными стандартами, называемыми *протоколами.* Они определяют характер аппаратного взаимодействия компонентов сети (*аппаратные протоколы*) и характер взаимодействия программ и данных (*программные протоколы*). Физическую поддержку протоколов выполняют устройства, называемые интерфейсами, и программные средства — программы поддержки протоколов.

Единый протокол передачи данных по сети Интернет TCP/IP (с 1983 г. для сети NSFNet — сеть Национального научного фонда США) обеспечивает устойчивость Глобальной сети. *Транспортный протокол TCP* (Transmission Control Protocol) разбивает файлы на объекты при передаче и собирает их при получении. *Адресный протокол IP* (Internet Protocol) — протокол маршрутизации, обеспечивающий доставку. Если отправляется письмо, то пишутся адреса отправителя и получателя. Информация упаковывается в конверт, содержание которого называется IP-пакетом, и «пишутся» IP-адреса получателя. Например, «Кому: 198.78.213.185»; «От кого: 193.124.5.31». По наборам цифр можно определить адресата.

В соответствии с используемыми протоколами компьютерные сети принято разделять на *локальные* (*LAN — Local Area Network*) и *глобальные* (*WAN — Wide Area Network*). По территориальному признаку локальные сети отличаются компактностью. Глобальные сети имеют, как правило, увеличенные географиче-

ские размеры и создаются для формирования единого мирового информационного пространства (например, Интернет — межсеть, сеть сетей — Всемирная компьютерная сеть).

Сети, объединяющие компьютеры в пределах одного региона, называются *региональными*. *Корпоративные* сети создаются организациями, заинтересованными в защите своей информации (военные, банковские и др.).

Группы сотрудников, работающих над одним проектом в рамках одной локальной сети, называются *рабочими группами*.

Совокупность приемов разделения и ограничения прав участников компьютерной сети называется *политикой сети*. Управление сетевыми политиками называется *администрированием сети*. Лицо, управляющее организацией работы участников локальной компьютерной сети, называется *системным администратором*.

Локальные сети — совокупность ЭВМ и линий связи, расположенных на небольшой территории и принадлежащих, как правило, одной организации. В локальных сетях используются качественные линии связи: коаксиальные кабели, оптоволоконные кабели, витая пара. По типу используемых компьютеров локальные сети относятся к однородным, т. е. к программно-совместимым.

Глобальные сети — территориально рассредоточенные на большие расстояния компьютеры, объединенные скоростными каналами связи. В глобальных сетях применяются сложные методы передачи данных: модуляция, асинхронность, контрольное суммирование, квитирование, повторная передача искаженных фрагментов. Скорость передачи данных — 2400, 9600, 28 800, 33 600 бит/с и 64 Кбит/с, на магистральных каналах — до 2—10 Мбит/с. Но в некоторых коммерческих глобальных сетях, использующих оптическую цифровую передачу данных по оптоволоконным линиям связи, скорости передачи данных приближаются к скоростям передачи по локальным сетям. По типу используемых компьютеров глобальные сети относятся к неоднородным, т. е. к программно-несовместимым.

Корпоративные сети — компьютерные сети, основной задачей которых является передача, защита и использование информации в рамках определенной сферы деятельности.

Региональные сети — компьютерные сети, предназначенные для обслуживания информационными ресурсами крупной территории. Они используют цифровые магистральные линии связи

(часто оптоволоконные), скорость передачи информации составляет примерно 45 Мбит/с. Часто объединяют локальные сети территории и соединяют их с глобальной сетью.

В зависимости от способа управления сетевыми ресурсами сети могут быть централизованными (управление сосредоточено на одном из серверов) и децентрализованными (управление осуществляет каждый сервер).

В настоящее время широко применяются многотерминальные локальные и глобальные системы. В рамках таких систем успешно решаются задачи доступа к ЭВМ-серверам и обмена информацией между различными пользователями. Видеотерминалы соединяются с компьютерами через телефонные сети с помощью специальных устройств — модемов. Модем обеспечивает модуляцию и демодуляцию сигнала при его передаче по телефонным линиям. Основная характеристика качества модема — скорость передачи (пропускная способность канала), измеряемая в битах в секунду (бодах). Имеются протоколы модемной связи, которые утверждает Международный телекоммуникационный союз.

Принцип удаленного доступа позволяет обмениваться информацией между различными ЭВМ в автоматическом режиме, т. е. осуществлять обмен файлами, синхронизировать БД, передавать сообщения по электронной почте и т. д.

Потребность в совместном использовании информационных ресурсов, их сборе и передаче привела к соединению мини-компьютеров (персональных компьютеров) в локальные вычислительные сети. В локальных сетях уже в 1980-х годах широко применялись стандартные технологии объединения компьютеров в сеть (Ethernet, ARC net и др.). Для создания сети нужно было иметь сетевые адаптеры и стандартный кабель, к которому подключались адаптеры через разъемы. На компьютере в этом случае устанавливается одна из сетевых операционных систем (например, NetWare и т. п.).

Локальная сеть позволяет совместно использовать периферийные устройства. Если к сети подключено более 10 компьютеров, то для увеличения производительности и надежности хранения информации в сети некоторые компьютеры специально выделяют для хранения файлов или программ-приложений. Эти компьютеры называются *серверами*.

Выделяют четыре группы локальных сетей (по признакам расстояния и скорости передачи информации).

1. ЛВС (локальные вычислительные сети), ориентированные на массового пользователя. Обеспечивают передачу информации на расстояние 100—500 м со скоростью передачи информации по локальной сети υ = 2400—19 200 бод (1 бод равен 16 бит/с). ЛВС, кроме ПЭВМ объединяющих микропроцессорную технику, встроенную в технологическое оборудование (средства автоматизации, кассовые аппараты и др.), а также средства электронной почты, обеспечивают передачу информации на расстояние до 1 км со скоростью υ = 19 200 бод — 1 Мбод.

2. ЛВС, объединяющие ПЭВМ, мини-ЭВМ и ЭВМ среднего класса. Служат для организации управления сложными производственными процессами: гибких автоматизированных мест (ГАМ), систем автоматизированного проектирования (САПР) и др. Обеспечивают передачу информации на расстояние до нескольких километров со скоростью υ = 120 Мбод.

3. ЛВС, объединяющие все классы ЭВМ. Служат для управления сложным производством, отраслью. Обеспечивают передачу информации на расстояние до 10 км со скоростью υ = = 10—50 Мбод.

Локальные сети постоянно совершенствуются. В них теперь используется новое коммуникационное оборудование: коммутаторы, шлюзы, маршрутизаторы.

Эта техника используется уже и для построения больших корпоративных сетей.

Кроме персональных компьютеров в сетях используют и другие типы ЭВМ, особенно большие ЭВМ, так называемые мэйнфреймы и супер-ЭВМ.

Появление высокоскоростных каналов связи привело к развитию глобальных сетей. С 1999 г. реализуется международный проект 174 стран по созданию сверхскоростных каналов связи протяженностью 275 тыс. км.

Глобальная сеть позволяет совместно использовать огромные информационные ресурсы. Например, к глобальной сети Интернет подключены десятки миллионов компьютеров, хранящих сотни миллионов файлов, документов и т. п. Сотни миллионов людей пользуются услугами Сети. Основу Интернета составляют более 100 млн серверов, постоянно подключенных к Сети. Подключиться к серверам можно с помощью локальных сетей или телефонных линий. Для подключения к Интернету нужно вы-

звать программу установки связи и затем ввести имя пользователя и пароль.

По существу, глобальные сети объединяют компьютеры всего мира. Глобальные сети используют уже существующие системы телефонной и телеграфной связи. Скорости передачи данных в таких сетях невысоки (десятки килобит в секунду), а передача файлов идет в фоновом режиме, обычно по электронной почте. Для правильной передачи данных применяют сложные процедуры контроля и восстановления данных.

Самая популярная в настоящее время глобальная сеть — Интернет, объединяющий локальные, региональные и другие сети в единое информационное пространство. Интернет включает самые разные модели ЭВМ, поэтому для их общения между собой приняты протоколы.

Протокол — совокупность правил, определяющих передачу данных между компонентами компьютерной сети.

В соответствии с IP-протоколом передаваемые данные разбиваются с помощью прикладной программы на блоки определенного формата, упаковываются в пакеты, имеющие номер и заголовок, и передаются по определенным маршрутам, а поступающие пакеты обрабатываются. TCP — протокол управления потоком данных, следящий за комплектностью и порядком получения и сборки пакетов. Работа по этим протоколам подобна работе почтовой службы.

На основе протоколов IP и TCP разработаны сетевые сервисные протоколы:

- FTP (File Transfer Protocol) — протокол передачи файлов;
- HTTP (Hyper Text Transfer Protocol) — протокол передачи гипертекста;
- SMTP (Simple Mail Transfer Protocol) — протокол пересылки электронной почты;
- NNTP (Network News Transfer Protocol) — протокол передачи новостей (телеконференций);
- Telnet — протокол удаленного доступа.

В Интернете в десятках миллионов компьютеров хранится огромный объем информации, по существу, всем доступной. Компьютер, подключенный к Интернету, имеет свой адрес, по которому его можно найти с любого другого компьютера сети.

В сетях широко применяются методы обработки всех видов информации, в том числе аудио- и видеоинформации, т. е. мультимедийной информации.

Синтез технологий всех видов информационных сетей (локальных, глобальных, вычислительных, телефонных, телевизионных и т. д.), над которым сейчас работают специалисты, позволит создать совершенную технологию передачи данных. Для создания такой технологии постоянно совершенствуются все элементы сетей: компьютеры, коммуникационное оборудование, операционные системы и сетевые приложения.

Компьютерные сети по функциональному признаку делят на следующие системы:

- абонентские (компьютер в составе сети, обеспечивающий пользователю доступ к ее ресурсам);
- коммутационные (узлы соединения (коммутации) сети передачи данных между абонентскими системами — процессоры телеобработки или специальные сетевые процессоры);
- хост-системы (сетевые серверы — специальные компьютеры, выполняющие основные функции управления сетью, сбора, обработки, хранения и предоставления информации пользователям сети). По предоставлению сервисных услуг выделяют файл-серверы, серверы доступа и т. д.

При подключении локальной сети к Интернету каждый компьютер этой сети через свой хост-компьютер получает возможность выхода в глобальную сеть.

Способы подключения компьютеров к сети следующие.

Каждый компьютер, подключенный к сети, имеет специальную плату (сетевой адаптер). Соединение компьютеров осуществляется с помощью кабелей. Основная функция адаптера — передача и прием информации из сети.

Подключиться к сети Интернет можно через поставщика услуг Интернета (провайдера). Эта организация осуществляет коммерческое подключение к своему узлу и выделяет IP-адреса. Подключение может осуществляться по *каналам выделенным* или *коммутируемым*. Организации, передающие большие объемы информации, используют выделенные каналы. Для этого прокладывают новую или арендуют готовую линию связи (кабельную, оптоволоконную, радиоканал, спутниковый канал и др.). Коммутируемое соединение — временное, не требует специальной линии связи и может осуществляться по телефонной линии (аналоговой или цифровой). Подключение (коммутацию) автоматически выполняет АТС. Но пропускная способность этих линий низкая: 50 Кбит/с на аналоговых линиях и 60—120 Кбит/с на

цифровых линиях. Скорость передачи информации (пропускная способность) по каналу является основной его характеристикой. Скорость передачи информации по локальной сети — 10—100 Мбит/с. Пропускная способность линий связи глобальной сети Интернет— 1—10 Гбит/с.

ОС Windows позволяет осуществить разные виды сетевых подключений: прямое соединение с соседним компьютером, подключение к локальной сети, удаленный доступ к Интернету. Чтобы подключиться к компьютеру провайдера, нужно выбрать команды **Пуск → Настройка → Сетевые подключения → Мастер новых подключений**. Поставщик услуг должен сообщить следующие данные: номер телефона, по которому производится соединение; имя пользователя (Login); пароль (Password); IP-адрес сервера доменной службы DNS (см. далее).

Работу глобальной сети обеспечивает *коммуникационное оборудование* (шлюзы, мосты, маршрутизаторы, концентраторы, модемы, кабельные соединения).

Мост — устройство для соединения между собой двух локальных сетей, использующих одинаковые протоколы. В качестве моста может быть использован специализированный компьютер или программное средство. Мост не позволяет выпускать за пределы локальной сети данные для внутреннего пользования.

Шлюзы — специальные средства, используемые для связи между собой нескольких локальных сетей, работающих по разным протоколам. Шлюзы могут быть как аппаратными, так и программными.

Маршрутизаторы — специальные средства (специальные компьютер или программа, работающая на узловом сервере сети), занимающиеся вопросами доставки информации на компьютеры, находящиеся «ближе» или «дальше». «Ближе» — означает лучшие условия связи и пропускную способность, а не географическое положение.

Концентратор — сетевое устройство для объединения нескольких устройств Ethernet в общий сегмент сети. Устройства подключаются с помощью витой пары, коаксиального кабеля или оптоволокна.

Модемы обеспечивают модуляцию и демодуляцию сигнала при его передаче по телефонным линиям (см. раздел 2.5). Основная характеристика качества модема — скорость передачи информации по линии канала связи (как правило, $v = 33{,}6$ и 56 Кбит/с). Имеются протоколы модемной связи, например

v.34+ и v.90, которые утверждает Международный телекоммуникационный союз.

Существует два способа подключения к сети: через внешний модем (при этом обеспечивается скорость $v \geq 28\,800$ бит/с); через внутренний (ISDN-карта) модем (при этом обеспечивается $v \geq 24\,000$ бит/с);

Кабельные соединения компьютеров осуществляются с помощью кабелей различного типа (коаксиального, витой пары, оптоволоконного). По топологии и архитектуре различаются следующие сети: линейная шина (на базе последовательного соединения), звезда (каждый компьютер соединен с сервером отдельным кабелем) и др.

Чтобы компьютеры могли найти друг друга, существует *единая система адресации*, использующая 32-битный IP-адрес, который имеет каждый компьютер, подключенный к Интернету. Общее количество таких адресов можно подсчитать по формуле

$$N = 2^{32} = 4\,294\,967\,296,$$

т. е. более 4 млрд адресов.

IP-адрес состоит из двух частей: адреса сети и адреса компьютера в данной сети. Сети разбиты на классы A, B, C:

- A — число от 0 до 127 (адрес в сети 7 бит, адрес компьютера 24 бит);
- B — число от 128 до 191 (адрес в сети 14 бит, адрес компьютера 16 бит);
- C — число от 192 до 223 (адрес в сети 21 бит, адрес компьютера 8 бит).

Серверы *службы имен доменов* (*DNS — Domain Name System*) переводят доменные имена в связанные с ними IP-адреса. Запрос сначала обрабатывается сервером DNS, а затем отправляется по IP-адресу. Деление Всемирной сети на *домены* по принципу принадлежности произошло во 2-й половине 1980-х годов, когда образовалась система доменных имен DNS. Доменная система имен ставит в соответствие числовому IP-адресу уникальное доменное имя разного уровня. Например, Россия имеет имя ru географического домена. Основной сервер компании Microsoft имеет имя www.microsoft.com (коммерческий домен).

При подключении локальной сети предприятия к глобальной сети важную роль играет понятие *сетевой безопасности*. В частности, должен быть ограничен доступ в локальную сеть для по-

сторонних лиц извне, а также ограничен выход за пределы локальной сети для сотрудников предприятия, не имеющих соответствующих прав. Для обеспечения сетевой безопасности между локальной и глобальной сетью устанавливают так называемые *брандмауэры*. Брандмауэром может быть специальный компьютер или компьютерная программа, препятствующая несанкционированному перемещению данных между сетями.

Информация и компьютер должны быть защищены от *вирусов*.

Существуют методы защиты от вирусов: программные (антивирусные программы); аппаратные (средства аппаратной защиты) и организационные (недопущение несанкционированного доступа, резервное копирование данных и хранение этих копий на другом компьютере или в сейфе на дисках и др.).

Одним из эффективных методов защиты информации является *шифрование* данных. Метод основан на использовании ключа, который закрывает доступ к прочтению информации. Прочитать ее можно, только зная ключ. Если для шифрования и чтения информации используют один и тот же ключ, то такой криптографический процесс называется *симметричным*. Если для шифрования и чтения информации используют два ключа — один открытый (публичный), второй закрытый (личный), — то такой криптографический процесс называется *асимметричным*. Ключи подобраны так, что сообщение, зашифрованное первым ключом, можно расшифровать вторым. Вопрос изучения и применения алгоритмов шифрования данных называется криптоанализом и законодательно регулируется во многих странах. В России разрешены средства шифрования, которые прошли государственную сертификацию в административных органах.

Для подтверждения подлинности сведений служит *электронная подпись*. При ее создании используется специальная программа, полученная, например, от банка, и с ее помощью создаются два ключа: публичный и закрытый (личный). Публичный ключ передается банку. Передаваемое от клиента сообщение банку кодируется этим ключом, а подпись клиента — закрытым ключом. Банк читает сообщение с помощью своего закрытого ключа, а подпись — с помощью публичного ключа клиента.

Все делопроизводство в Интернете выполняется с использованием несимметричного шифрования. Это в основном исключает несанкционированный доступ к документам.

Но и эта система имеет недостатки. Возникают трудности с регистрацией даты отправления сообщения, так как ее легко можно изменить средствами ОС. Поэтому она не имеет юридической силы. При необходимости нужно выполнить *сертификацию даты/времени* с участием третьей независимой стороны. Этот участник выполняет «приписку» точной даты и времени, зашифрованную его закрытым ключом. Работа автоматизированна. Разработаны также системы сертификации с помощью асимметричного шифрования веб-узлов, издателей (при распространении программного обеспечения через Интернет).

Проверку сертификатов можно осуществить с помощью браузеров, например, с помощью команд меню браузера MS Internet Explorer **Сервис → Свойства обозревателя → Содержание → Сертификаты → Доверенные корневые центры сертификации**.

В 1994 г. был создан Консорциум Всемирной паутины — W3C — организация, разрешающая и внедряющая *технологические стандарты для WWW*. Разработка стандартов проходит четыре стадии: рабочий проект, последний созыв, возможная рекомендация, предлагаемая рекомендация.

Полный список организаций, сотрудничающих с W3C, включает официальные сайты:

- Консорциума Всемирной паутины;
- Массачусетского института технологий;
- Европейского консорциума по исследованиям в области информатики и математики;
- университета Кейо.

Стандарты, утвержденные W3C:

- Annotea;
- CC/PP;
- Compound Document Formats;
- CSS;
- DOM;
- HTML;
- HTTP;
- lnkML;
- MathML;
- OWL;

- PICS;
- PNG;
- P3P;
- RDF;
- SMIL;
- SOAP/XMLP;
- SPARQL;
- Style;
- SVG;
- TAG;
- Timed Text;
- URI/URL;
- Voice Browser;
- WAI;
- WebCGM;
- Web Services;
- XForms;
- XHTML;
- XInclude;
- XLink;
- XML;
- XML Base;
- XML Binary Characterization;
- XML Encryption;
- XML Key Management;
- XML Query;
- XML Schema;
- XML Signature;
- XPath;
- XPointer;
- MXSL и XSLT.

Другие организации, разрабатывающие сетевые стандарты:

- Internet Society — ISOC;
- The Internet Engineering Task Force — IETF;

- International Organization for Standardization — ISO (ИСО — Международная организация по стандартизации, создана в 1946 г.);
- Web Standards Group — WSG;
- The Web Standards Project;
- Unicode Organization;
- The Semantic Web Community Portal.

ИСО, например, разработала стандарт ISO/IEC 10026 — Open System Interconnection (Сетевая модель OSI, ACID), стандарт ISO 10646 — Unicode и др.

В сентябре 2005 г. Россия вошла в Совет ИСО.

Контрольные вопросы

1. Каково назначение компьютерных сетей?
2. Каковы типы ресурсов?
3. Каковы уровни архитектуры сетей по модели ISO/OSI?
4. Каковы назначение и основные функции глобальной компьютерной сети Интернет?
5. Какие виды сетей вы знаете?
6. Какова классификация локальных сетей?
7. Что такое единый протокол передачи данных по сети Интернет?
8. Что включает коммуникационное оборудование глобальной сети?
9. Что такое единая система адресации компьютеров в сети Интернет?
10. Каковы функции службы DNS?
11. Что используют для обеспечения сетевой безопасности?
12. Для чего предназначена электронная подпись и как она создается?
13. Какие вы знаете стандарты, утвержденные для работы в WWW?

7.1.1. Работа в Интернете. Службы Интернета

Подключение к Интернету. С помощью команд **Пуск → Программы → Стандартные → Связь → Удаленный доступ к сети → Провайдер → Интернет** вызвать диалоговое окно **Установка связи**. Ввести идентификатор (login) в поле **Имя пользователя** и пароль

(password) в поле **Пароль**. Нажать кнопку **Подключиться**. Должно появиться информационное окно, содержащее характеристики скорости, длительности и др.

Настройка Internet Explorer. Это специальная программа (браузер, обозреватель) для навигации по Интернету. Запуск ее можно выполнить разными способами: из главного меню, с панели задач, с рабочего стола. Открывшееся окно браузера содержит строку меню, панель инструментов **Кнопки**, адресную строку и рабочую область для просмотра веб-страниц. Для просмотра страниц в кодировке Windows необходимо выполнить команды **Вид → Вид кодировки → Кириллица**.

С помощью команд **Сервис → Свойства обозревателя** вызывается диалоговое окно **Свойства обозревателя**, в котором следует выбрать **Общие Домашняя страница**. В поле домашней страницы нужно ввести с клавиатуры нужный URL-адрес — универсальный указатель ресурсов URL (Universal Resource Locator), включающий способ доступа к документу, имя сервера, на котором находится документ, а также путь к файлу (документу). Например, адрес веб-сайта «Информатика 2000» будет следующим: http://schools.keldysh.ru/info2000/index.htm, где http:// — протокол доступа (протокол передачи гипертекста — Hyper Text Transfer Protocol), schools.keldysh.ru — имя сервера, /info2000/index.htm — путь к файлу веб-страницы, которая входит в состав веб-сайта. Нажав кнопку **Домой**, можно загрузить начальную страницу этого веб-сайта.

Поиск информации в Интернете можно рассмотреть на примере российского поискового сервера Rambler, его адрес: http://www.rambler.ru. Начальная страница его поисковой системы содержит *поле поиска* и *каталог* (список разделов). В поле поиска следует ввести ключевые слова для поиска определенных документов. Появляется список сайтов и веб-страниц (в порядке рейтинга), в которых есть эти ключевые слова. Для документов приводятся ссылка, имя, адрес, ключевые слова и описание. Щелчок по подчеркнутой ссылке приведет к загрузке титульной страницы сайта.

Много файловых архивов хранится на так называемых FTP-серверах. Для доступа к ним используется специальный протокол передачи файлов FTP (File Transfer Protocol).

Для *интерактивного общения в Интернете* имеется специальное программное обеспечение. Общение может быть реализова-

но в форме разговора, аудио- или видеоконференций. Наиболее распространенной программой является NetMeeting, входящая в состав Internet Explorer. После ее загрузки нужно в поле **Сервер** из списка выбрать сервер, на котором будет происходить общение. В главном окне будет представлен список участников общения с некоторыми сведениями о них. Для присоединения к разговору нужно ввести команды **Сервис → Разговор (Chat)**. В поле **Сообщение** окна **Разговор** следует ввести текст сообщения. В поле **Отправить** нужно из списка выбрать получателей сообщения. Для всех участников общения при выборе команды **Сервис → Доска** может быть предоставлена *Доска*, которая позволяет всем участникам создавать и редактировать общий рисунок. Участники могут также коллективно работать с приложениями MS Word (например, с помощью команд **Сервис → Общие приложения → Текстовый редактор**), Excel и др.

Для определения текущего IP-адреса своего компьютера нужно в окне **Сеанс MS-DOS** ввести команду `winipcfg`. В появившемся диалоговом окне **Конфигурация IP** можно увидеть полную информацию о параметрах текущего подключения к Интернету, включая IP-адрес вашего компьютера.

Упрощенно *служба Интернета* — это пара групп программ, взаимодействующих между собой по определенным правилам — протоколам. Одна программа — сервер, вторая — клиент. Разные службы работают по разным прикладным протоколам.

Telnet — служба удаленного управления компьютером. По протоколу этой службы можно подключиться к удаленному компьютеру и управлять им (консольное или терминальное управление). Применяют для дистанционного управления техническими объектами и др. Простейший клиент Telnet имеется в ОС Windows XP (файл telnet.exe).

Передачи файлов осуществляются *службой FTP*, которая основывается на виртуальном соединении сетевого уровня и соответствующих ему протоколах FTP. Для получения из Интернета файла необходимо на ПК иметь программу, являющуюся FTP-клиентом, и установить связь с FTP-сервером, оказывающим услуги FTP. Имеются десятки тысяч FTP-серверов с анонимным доступом. В этом случае в качестве имени пользователя следует ввести слово anonymous, а в качестве пароля — адрес электронной почты.

Списки рассылки (*Mail list*). Когда партнеров по обмену информацией нет, то можно подписаться на списки рассылки для получения разнообразной почтовой информации. Mail list — специальные тематические серверы, собирающие информацию по темам и отсылающие ее в виде сообщений подписчикам.

Телеконференции (*Usenet*) — совещания «на расстоянии», когда люди обсуждают те или иные вопросы, видя и слыша друг друга. В Интернете имеются десятки тысяч конференций, где обсуждаются всевозможные проблемы. Каждой конференции выделяется свой почтовый ящик на серверах сети, которые поддерживают работу определенной телеконференции. Пользователи могут посылать туда свои сообщения. Задать вопрос можно, обращаясь ко всему миру. Получить ответ или совет можно также от многих. Многие специалисты постоянно просматривают группы новостей, касающиеся их сферы деятельности, т. е. осуществляют мониторинг информации. Серверы периодически обмениваются информацией ящиков телеконференций. Бесполезная информация фильтруется, т. е. модерируется. Модератором может быть программа, фильтрующая сообщения по ключевым словам (автоматическая модерация). Для работы на телеконференциях существуют специальные клиентские программы, например MS Outlook Express и др.

Электронная почта (E-mail) — наиболее распространенный сервис Интернета (подробно см. раздел 7.3). Эта служба основана на двух прикладных протоколах: SMTP — для отправки корреспонденции с компьютера на сервер, POP3 — для приема поступивших сообщений. Адрес электронной почты состоит из двух частей: *имя пользователя@имя сервера* (на котором пользователь размещает свой почтовый ящик, доменное имя). Свое имя задает пользователь. Расширение доменного имени (после точки) означает принадлежность его к определенному типу организаций. Например, .gov — правительственные учреждения, .com — коммерческие организации и т. д.

Для работы с электронной почтой имеются специальные почтовые программы: MS Outlook Express, входящая в коммуникационный пакет MS Internet Explorer; Netscape Messenger, входящая в коммуникационный пакет Netscape Communication; The Bat! и Eudora Pro.

Word Wide Web (*WWW*) — Всемирная паутина — единое информационное пространство, включающее сотни миллионов взаимосвязанных электронных документов, хранящихся на де-

сятках миллионах серверах Интернета (веб-серверах). Документ Всемирной паутины называется *веб-страницей*. Группа тематически связанных веб-страниц называется *веб-сайтом* (веб-узлом, сайтом). Веб-документы не имеют жесткого форматирования. Создание веб-страниц выполняется с помощью языка разметки гипертекста HTML (см. раздел 7.2). В настоящее время в веб-документах применяют активные компоненты — объекты, содержащие программный код, который может исполняться. Браузеры следят, чтобы такой программный код не был для чужого компьютера вредоносным.

Программы для просмотра веб-страниц называются браузерами. Совокупность огромного числа гипертекстовых электронных документов, хранящихся на серверах Всемирной паутины, составляют гиперпространство документов, между которыми возможно перемещение, называемое веб-серфингом (ознакомительный просмотр) и веб-навигацией (поиск нужной информации). Каждый файл в этом пространстве имеет свое уникальное имя. Адрес его определяется *унифицированным указателем ресурса — URL*, состоящим из трех частей: указания имени протокола службы доступа к ресурсу (например, для службы WWW это HTTP — протокол передачи гипертекста: http://); указания доменного имени компьютера (сервера), на котором хранится ресурс (http://www.microsoft.com/); указания полного пути доступа к файлу на данном компьютере (http://www.microsoft.com/Files/New/info2008/index.htm). Следует соблюдать регистр символов, так как в Интернете символы в разных регистрах имеют разное значение.

ICQ (акроним I seek you — я ищу тебя) — служба мгновенного обмена сообщениями. Если компьютеры работают в Сети, то нужно лишь знать сетевой IP-адрес компьютера человека, с которым намерен общаться. Пользователю этой службы надо зарегистрироваться на ее центральном сервере http://www.icq.com и получить персональный идентификационный номер UIN (Universal Internet Number). Этот номер должны знать партнеры по общению.

IP-адрес может быть постоянным или динамическим, т. е. выдаваться только на сеанс подключения к Интернету.

IRC-сервис (*Internet Relay Chat*) — служба для прямого общения нескольких человек в режиме реального времени. Общение происходит в пределах канала связи. Существует несколько

клиентских программ для работы с IRC-сервисом, например mIRC.exe.

Служба каталогов. Один физический сервер может содержать много веб-узлов. Каждому веб-узлу отводится отдельный каталог на жестком диске сервера.

Поисковые указатели выполняют автоматическую каталогизацию информационных ресурсов. Указатели заняты поиском веб-ресурсов по ключевым полям. Поисковые указатели для обработки запросов пользователей применяют разные информационные технологии. Поисковый указатель работает в три этапа.

1. Поисковый указатель собирает информацию из Всемирной паутины и копирует ее на сервер поискового указателя. Для этого используют специальные программы, аналогичные браузерам. Они способны скопировать заданную веб-страницу на сервер поискового указателя, просмотреть ее, найти все гиперссылки, которые на ней имеются; ресурсы, которые найдены там; снова разыскать имеющиеся в них гиперссылки и т. д. Такие программы называют червяками, пауками, гусеницами, краулерами, спайдерами и др. Каждый поисковый указатель эксплуатирует для этой цели свою уникальную программу, которую нередко сам и разрабатывает.

2. Индексация. В ходе индексации создаются специальные базы данных, с помощью которых можно установить, где и когда в Интернете встречалось то или иное слово. Индексированная БД — это своего рода словарь. Она необходима для того, чтобы поисковая система могла очень быстро отвечать на запросы пользователей. Современные системы способны выдавать ответы за доли секунды, но если не подготовить индексы заранее, то обработка одного запроса будет продолжаться часами.

3. Обработка запроса клиента и выдача ему результатов поиска в виде списка гиперссылок. Например, пользователь хочет узнать, где имеются веб-страницы, на которых упоминается Дж. Бэкус. Он вводит слово «Бэкус» в поле набора ключевых слов и нажимает кнопку **Найти (Search)**. По своим базам указателей поисковая система в доли секунды разыскивает подходящие веб-ресурсы и формирует страницу результатов поиска, на которой рекомендации представлены в виде гиперссылок. Далее можно пользоваться этими ссылками для перехода к интересующим ресурсам.

сятках миллионах серверах Интернета (веб-серверах). Документ Всемирной паутины называется *веб-страницей*. Группа тематически связанных веб-страниц называется *веб-сайтом* (веб-узлом, сайтом). Веб-документы не имеют жесткого форматирования. Создание веб-страниц выполняется с помощью языка разметки гипертекста HTML (см. раздел 7.2). В настоящее время в веб-документах применяют активные компоненты — объекты, содержащие программный код, который может исполняться. Браузеры следят, чтобы такой программный код не был для чужого компьютера вредоносным.

Программы для просмотра веб-страниц называются браузерами. Совокупность огромного числа гипертекстовых электронных документов, хранящихся на серверах Всемирной паутины, составляют гиперпространство документов, между которыми возможно перемещение, называемое веб-серфингом (ознакомительный просмотр) и веб-навигацией (поиск нужной информации). Каждый файл в этом пространстве имеет свое уникальное имя. Адрес его определяется *унифицированным указателем ресурса — URL*, состоящим из трех частей: указания имени протокола службы доступа к ресурсу (например, для службы WWW это HTTP — протокол передачи гипертекста: http://); указания доменного имени компьютера (сервера), на котором хранится ресурс (http://www.microsoft.com/); указания полного пути доступа к файлу на данном компьютере (http://www.microsoft.com/Files/New/info2008/index.htm). Следует соблюдать регистр символов, так как в Интернете символы в разных регистрах имеют разное значение.

ICQ (акроним I seek you — я ищу тебя) — служба мгновенного обмена сообщениями. Если компьютеры работают в Сети, то нужно лишь знать сетевой IP-адрес компьютера человека, с которым намерен общаться. Пользователю этой службы надо зарегистрироваться на ее центральном сервере http://www.icq.com и получить персональный идентификационный номер UIN (Universal Internet Number). Этот номер должны знать партнеры по общению.

IP-адрес может быть постоянным или динамическим, т. е. выдаваться только на сеанс подключения к Интернету.

IRC-сервис (*Internet Relay Chat*) — служба для прямого общения нескольких человек в режиме реального времени. Общение происходит в пределах канала связи. Существует несколько

клиентских программ для работы с IRC-сервисом, например mIRC.exe.

Служба каталогов. Один физический сервер может содержать много веб-узлов. Каждому веб-узлу отводится отдельный каталог на жестком диске сервера.

Поисковые указатели выполняют автоматическую каталогизацию информационных ресурсов. Указатели заняты поиском веб-ресурсов по ключевым полям. Поисковые указатели для обработки запросов пользователей применяют разные информационные технологии. Поисковый указатель работает в три этапа.

1. Поисковый указатель собирает информацию из Всемирной паутины и копирует ее на сервер поискового указателя. Для этого используют специальные программы, аналогичные браузерам. Они способны скопировать заданную веб-страницу на сервер поискового указателя, просмотреть ее, найти все гиперссылки, которые на ней имеются; ресурсы, которые найдены там; снова разыскать имеющиеся в них гиперссылки и т. д. Такие программы называют червяками, пауками, гусеницами, краулерами, спайдерами и др. Каждый поисковый указатель эксплуатирует для этой цели свою уникальную программу, которую нередко сам и разрабатывает.

2. Индексация. В ходе индексации создаются специальные базы данных, с помощью которых можно установить, где и когда в Интернете встречалось то или иное слово. Индексированная БД — это своего рода словарь. Она необходима для того, чтобы поисковая система могла очень быстро отвечать на запросы пользователей. Современные системы способны выдавать ответы за доли секунды, но если не подготовить индексы заранее, то обработка одного запроса будет продолжаться часами.

3. Обработка запроса клиента и выдача ему результатов поиска в виде списка гиперссылок. Например, пользователь хочет узнать, где имеются веб-страницы, на которых упоминается Дж. Бэкус. Он вводит слово «Бэкус» в поле набора ключевых слов и нажимает кнопку **Найти (Search)**. По своим базам указателей поисковая система в доли секунды разыскивает подходящие веб-ресурсы и формирует страницу результатов поиска, на которой рекомендации представлены в виде гиперссылок. Далее можно пользоваться этими ссылками для перехода к интересующим ресурсам.

Основная проблема современного Интернета связана с изобилием веб-страниц. Достаточно ввести в поле поиска простое слово — и поисковая система выдаст сотни тысяч ссылок, сгруппировав их по 10—20 штук на отображаемой странице. Как правило, пользователи просматривают не более 50 ссылок, стоящих первыми, и их интересует качество этих ссылок. Поэтому лучшие поисковые системы проявляют чудеса искусственного интеллекта в попытке отсортировать найденные ссылки по качеству их ресурсов.

При выдаче результатов поиска каждая поисковая система начинает проявлять свои лучшие (или худшие) индивидуальные черты. Операция сортировки полученных результатов называется ранжированием. Каждой найденной веб-странице система присваивает рейтинг, отражающий качество материала.

Высокие рейтинги получают веб-страницы, у которых ключевое слово, использованное в запросе, входит в заголовок. Уровень рейтинга повышается, если это слово встречается на веб-странице несколько раз, но не слишком часто. Благоприятно влияет на рейтинг вхождение нужного слова в первые 5 абзацев текста — они считаются самыми важными при индексации.

Очень хорошо, если ключевые слова, использованные в запросе, входят в альтернативный текст, сопровождающий иллюстрации. Для поисковой системы это верный признак того, что данная страница точно соответствует запросу.

Еще одним признаком качества является тот факт, что на страницу есть ссылки с каких-то других веб-страниц. Чем их больше, тем лучше — значит, страница популярна и обладает высоким показателем цитирования. Хорошая, грамотная работа веб-мастера способна значительно поднять посещаемость страниц. Если используют «хитрости», то поисковая система может и наказать веб-страницу, присвоив ей штрафной отрицательный рейтинг.

Автоматическая каталогизация поисковых каталогов основана на SMART-технологиях. Наиболее перспективной является модель векторного информационного пространства. Комбинируя термины и их весовые коэффициенты в различных областях, строят многомерные системы координат, в которых различные области знания описываются разными многомерными векторами. Для каждой веб-страницы поисковая система может построить математический вектор, основанный на формальном анализе

содержания, и отнести документ к определенным категории и разделу. Поэтому для документа достаточно хранить его URL-адрес и число, соответствующее вектору.

Работу службы WWW для пользователя обеспечивают веб-браузеры. Одна из самых популярных в настоящее время программ для просмотра документов WWW-серверов — Internet Explorer, включенная в ОС Windows. Используется для навигации по сети Интернет. Эта программа обеспечивает доступ к документам локального компьютера, ресурсам корпоративной сети Интранет и к информации Интернета. Она обеспечивает работу со Всемирной паутиной, предоставляет средства работы с локальными папками компьютера и файловыми архивами FTP, доступ к средствам связи через Интернет. Для запуска Internet Explorer можно использовать его ярлык на рабочем столе или на панели быстрого запуска, а также главное меню (**Пуск → Программы → Internet Explorer**). Работая с навигатором, можно просматривать веб-страницы, работать с гиперссылками, выделяемыми обычно синим цветом и подчеркиванием. В окне Internet Explorer отображается HTML-документ, полученный из Сети. HTML (Hyper Text Markup Language) — язык разметки гипертекста, специальный язык. Текст в нем кодируется в ASCII и поэтому может быть создан и отредактирован в любом редакторе. Перемещение (навигация) по HTML-документу осуществляется с помощью ссылок — выделенных в документе фраз или областей. При наведении на них указатель мыши принимает вид кисти руки, показывая тем самым, что это гиперссылка. Имеется меню и адресная строка. Во время просмотра веб-документов удобно пользоваться кнопками панели инструментов **Обычные кнопки (Веб-компоненты)**. Имеется возможность работы с несколькими окнами.

Internet Explorer 8.0 имеет собственные средства поиска. Задание на поиск можно дать, введя ключевые слова в специальном поле поиска и нажав кнопку **Найти (Find)**. Результаты поиска отображаются в виде списка ссылок.

Поисковые системы предназначены для поиска необходимой пользователям информации. Поиск может быть простым (по ключевым словам), расширенным (по ключевым словам, связанным операторами логических отношений), контекстным (по точной фразе), специальным (по ссылкам на адреса URL и данным в служебных полях).

В России пользуются поисковыми системами Яндекс (www.yandex.ru), Rambler (www.rambler.ru), Googol (www.googol.ru). Исторически наиболее популярной поисковой системой является Rambler (см. раздел 1.1). Она начала работать раньше других и долгое время лидировала по размеру поискового указателя и качеству услуг поиска.

Но сегодня ее начинает обгонять система Яндекс, сочетающая возможности поискового указателя и поискового каталога. Поисковый указатель системы — примерно 60 млн веб-страниц на самые актуальные ресурсы. Система — лидер по уровню актуальности и имеет очень гибкий язык для расширенного поиска.

Поисковая система Northern Light (www.northernlight.com) предназначена для проведения научного поиска.

Самым большим поисковым указателем обладает поисковая система Fast Search (www.alltheweb.com).

Поисковые системы *реального времени* — новое направление в технологии поиска. Пользователь должен подключиться к центральному серверу службы, получить оттуда и установить на своем компьютере клиентскую программу. Эта программа подключается к браузеру и работает как дополнительная панель.

При каждом запуске браузера клиентская программа работает вместе со своим центральным сервером. Она передает серверу копии всех веб-страниц пользователя, выполняя функции автоматического *червя*. Но есть два существенных отличия:

- человек в ходе навигации руководствуется не теми же принципами, что автоматическая программа, поэтому сервер получает копии не всех веб-ресурсов, а тех, что заинтересовали кого-то из его клиентов;
- если поставкой веб-ресурсов занимаются несколько миллионов постоянных клиентов, индексация веб-пространства происходит быстрее.

Для пользователя важно, что, на какой бы странице он ни находился, система всегда готова предложить ему список веб-страниц, близких по тематике и содержанию.

Контрольные вопросы

1. Как подключиться к Интернету?
2. Что такое навигатор и как осуществить поиск информации в Интернете?

3. Какие службы Интернета вы знаете? Кратко охарактеризуйте их.
4. Что такое Всемирная паутина?
5. Каковы основные прикладные программы для использования услуг Интернета?

7.2. Создание веб-документов

Создание веб-сайтов осуществляют с помощью языка разметки гипертекстовых документов HTML. Технология состоит в том, что в обычный текстовый документ вставляются *управляющие символы* (*теги*). В результате получают веб-страницу. Документ может быть создан в любом текстовом редакторе (например, в Блокноте). Сначала нужно разработать проект сайта, определить, сколько в нем будет страниц и каких. Например, сайт будет содержать титульный лист «Самолеты» и четыре страницы: «Фирмы-изготовители», «Комплектующие», «Цены», «Покупатели».

1. Открыть окно текстового редактора Блокнот. Задать вид веб-страницы с помощью тегов, заключенных в угловые скобки. Пара тегов называется контейнером. Закрывающий тег должен содержать прямой слэш / перед обозначением. Веб-страница должна иметь заголовок и содержимое. *Заголовок*, содержащий название документа и не отображаемую справочную информацию о странице, нужно заключить в контейнер `<HEAD> </HEAD>`. Название веб-страницы должно быть в контейнере `<TITLE> </TITLE>` и отображаться в строке заголовка программы. Основное содержимое страницы нужно поместить в контейнер `<BODY> </BODY>`. Оно может содержать информацию разного вида (текст, рисунок и т. д.). Созданную титульную страницу следует сохранить в виде файла под именем index.htm или index.html. Целесообразно создать для сайта отдельную папку и сохранить в ней все созданные страницы.

2. Ввести HTML-код, тогда для титульной страницы в Блокноте будет следующий текст:

```
<HTML>
<HEAD>
<TITLE>Самолеты</TITLE>
</HEAD>
<BODY>
```

```
Необходимая информация для Вас —
Самолеты:
</BODY>
</HTML>
```

Текст можно отформатировать: заголовок выделить крупным шрифтом (<H1> — самый крупный, <H6> — самый мелкий), разместить его по центру страницы (атрибут ALIGN="center"), выровнять по правой или левой границе (ALIGN = "right" или ALIGN="left"), изменить цвет (например, <FONT COLOR="blue">. Заголовок следует отделить от содержимого горизонтальной линией с помощью тега <HR>.

3. В редакторе Блокнот в контейнер <BODY> вставить последовательность тегов:

```
<H1 ALIGN="center"
<FONT COLOR="blue">
Необходимая информация для Вас —
Самолеты:
</FONT>
</H1>
<HR>
```

4. Вставить изображение можно, используя тег <IMG> с атрибутом SRC="имя файла". Например, <IMG SRC="sam.gif">.

Рис. 7.1. Окончательный вид веб-страницы

Атрибут ALT позволяет задать поясняющий текст, а ALIGN — расположить его с любой заданной стороны от рисунка. Рисунок должен быть в одном из трех форматов: GIF, JPG, PNG. Если формат другой, то его переводят в один из трех указанных с помощью графического редактора (например, Photo Editor).

5. В редакторе Блокнот в контейнер <BODY> вставить следующие теги:

```
<IMG SRC="sam.gif">
ALT= "Страницы этого сайта позволяют вам больше узнать
     о самолетах..."
ALIGN="left">
```

Тогда результат просмотра нашей титульной страницы в программе должен быть следующим (рис. 7.1).

Контрольные вопросы

1. Что такое веб-сайт?
2. Каково назначение языка HTML?
3. Каковы основные характеристики языка HTML?
4. Как создать веб-документ?
5. Как вставить рисунок?

7.3. Создание сообщений для электронной почты

Электронная почта (*E-mail*) — один из самых распространенных видов сетевых услуг. При регистрации в сети Интернет каждый пользователь получает там свой почтовый ящик — подкаталог на диске узлового компьютера сети (сервера). С помощью специальной программы можно подготовить сколько угодно писем и положить их в папку для исходящих сообщений. Далее следует соединиться с компьютером — поставщиком услуг Интернет и обменяться почтой.

Для обмена письмами используется особая система адресации. В этой системе адреса записываются с помощью букв латинского алфавита и цифр в следующем формате: <имя пользователя>@<имя компьютера>. Слева от знака @ стоит имя пользователя, а справа — имя сервера, на котором находится

почтовый ящик. Имя компьютера состоит из нескольких слов или сокращений, разделенных точками. Любой компьютер в Интернете имеет свой IP-адрес в соответствии с доменной системой имен.

В ОС Windows имеется встроенное средство работы с электронной почтой — программа Outlook Express. Запуск программы осуществляется с помощью команд **Пуск** → **Программы** → **Outlook Express**. Появляется окно программы, состоящее из трех частей (рис. 7.2).

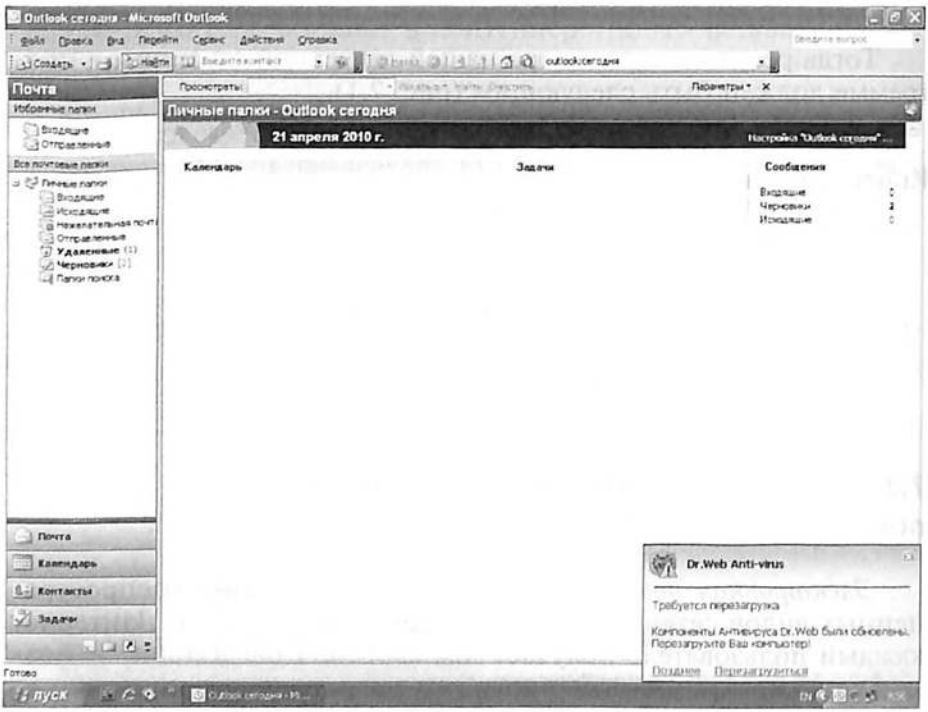


Рис. 7.2. Окно Outlook Express

Левая часть окна программы Outlook Express содержит перечень папок с корреспонденцией: Входящие, Исходящие, Отправленные, Удаленные, Черновики. В папках хранятся объекты разных типов. Правая часть окна разделена пополам. В верхней части показывается содержимое выделенной папки, в нижней — содержимое выделенного сообщения.

Для *настройки* работы почтовой программы с почтовым сервером сначала следует создать учетную запись Почта Интернета, с помощью которой можно отправлять и принимать электронную почту с почтового сервера. С помощью команд **Сервис → Учетные записи** открывают диалоговое окно **Учетные записи Интернета**. Далее следует открыть вкладку **Почта**, нажать кнопку **Добавить** и выбрать пункт **Почта**. В диалоговом окне **Мастер подключения к Интернету** в поле **Ваше имя** написать имя. В следующие окна следует последовательно переходить, нажимая кнопку **Далее**. В окне, где имеется поле **Адрес электронной почты**, указать адрес, заданный при регистрации у провайдера. В окне, где имеются поля **Сервер для входящей почты (POP3)** и **Сервер для исходящей почты (SMTP)**, указать имена серверов входящей и исходящей почты, полученные при регистрации у провайдера. В поле **Тип сервера для входящей почты** следует указать POP3 — протокол почтового отделения, часто используемый для электронной почты. В поле **Учетная запись POP** необходимо вписать имя, которое было указано перед значком @. В поле **Пароль** следует указать пароль, полученный при регистрации у провайдера. В поле **Имя учетной записи почты сети Интернет** нужно указать также способ соединения с Интернетом — **С помощью телефонной линии**. В диалоговом окне **Выбор модема** указать тип модема. Выбрать пункт **Использовать уже имеющееся удаленное соединение** и указать соединение (например, Провайдер Интернета). После нажатия кнопки **Готово** можно пользоваться электронной почтой.

Создать почтовое сообщение (рис. 7.3) можно с помощью команд **Сообщение → Создать**. В верхней части окна в поле **Кому** указать электронный адрес адресата, например forum- books@mail.ru. В поле **Копии** указать адреса получателей копии сообщения. В поле **Тема** указать тему сообщения (например, «Фото»). Далее следует ввести текст сообщения в отведенную для него область. К сообщению можно присоединить какой-либо файл. Для этого с помощью команд **Вставка → Вложение файла** нужно вызвать диалоговое окно **Вставка вложений** и выбрать файл, созданный в других приложениях (графический, звуковой и т. д.). Сформировав сообщение и нажав кнопку **Отправить**, можно поместить сообщение в папку Исходящие.

Чтобы *отправить сообщение* адресату, нужно подключиться к Интернету и нажать кнопку **Отправить**. Отправленные сообщения будут доставлены на почтовый сервер и одновременно по-

мещены в папку Отправленные на вашем компьютере. Почтовый сервер передаст через Интернет ваши сообщения на почтовый сервер получателей, и послания окажутся в почтовых ящиках адресатов.

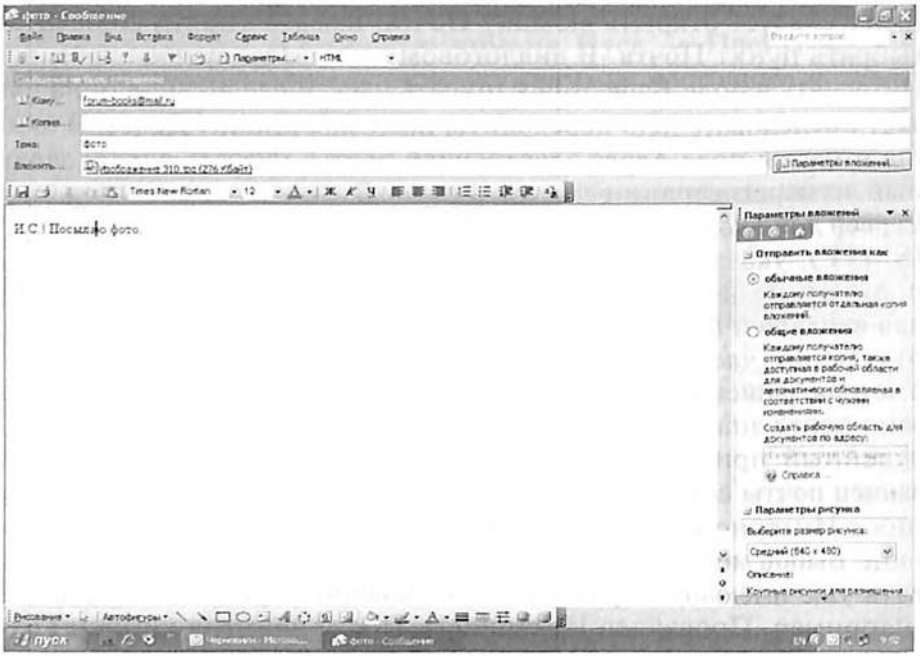


Рис. 7.3. Создание сообщения в Outlook Express

Для получения почты адресат должен установить соединение с Интернетом и выполнить операцию доставки почты с почтового сервера на свой компьютер. Сообщения будут размещены в папке Входящие.

Для работы с электронной почтой можно использовать веб-технологию. Имеются веб-сайты, которые предлагают бесплатно зарегистрировать свой почтовый ящик (например, по адресу http://mail.ru). В этом случае не требуются специальные почтовые программы, работу можно выполнять с помощью любого браузера после загрузки соответствующей веб-страницы, введя свой логин и пароль. Например, работая в системе Яндекс, можно использовать ее почту. На рис. 7.4—7.6 показана последовательность регистрации пользователя для получения почтового ящика и отправки сообщения по почте.

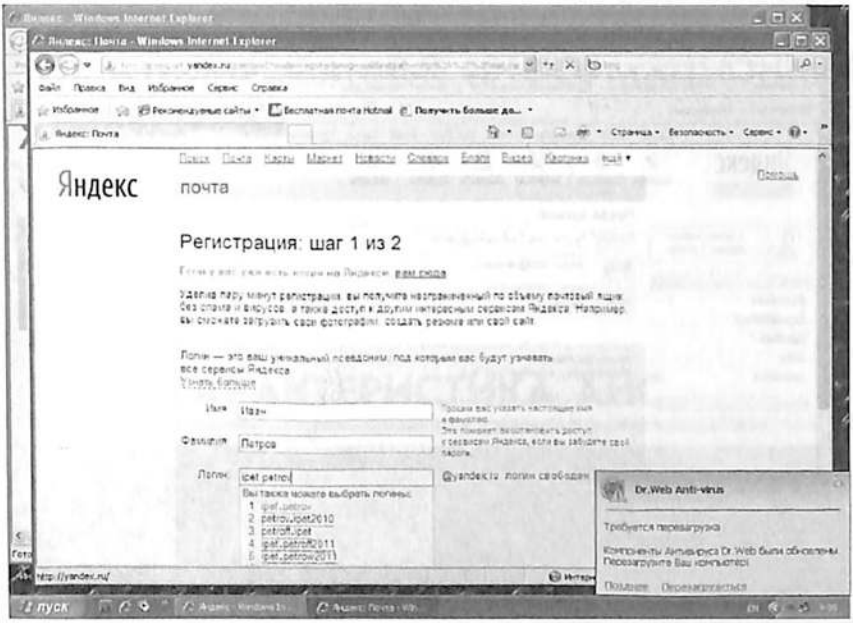


Рис. 7.4. Выбор логина для регистрации

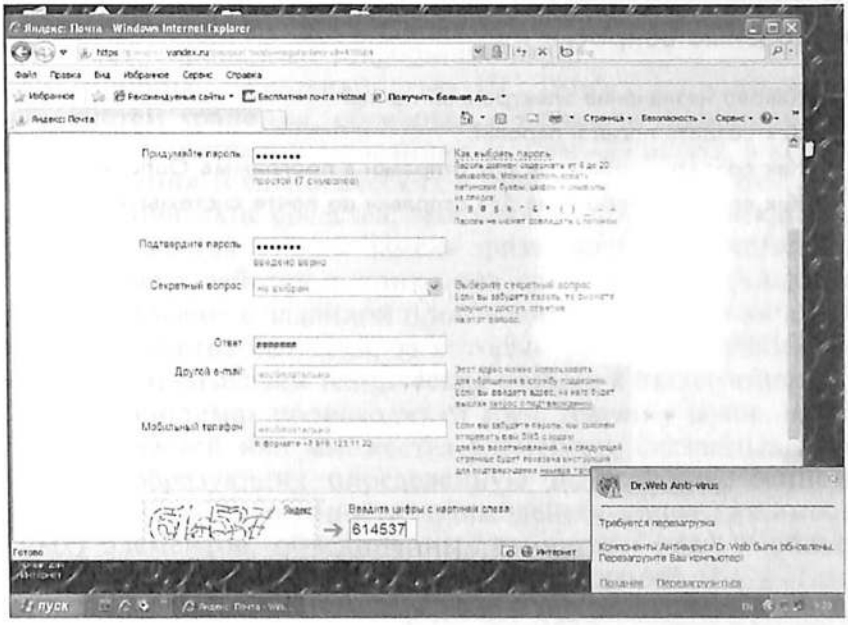


Рис. 7.5. Задание пароля и контрольного числа для регистрации

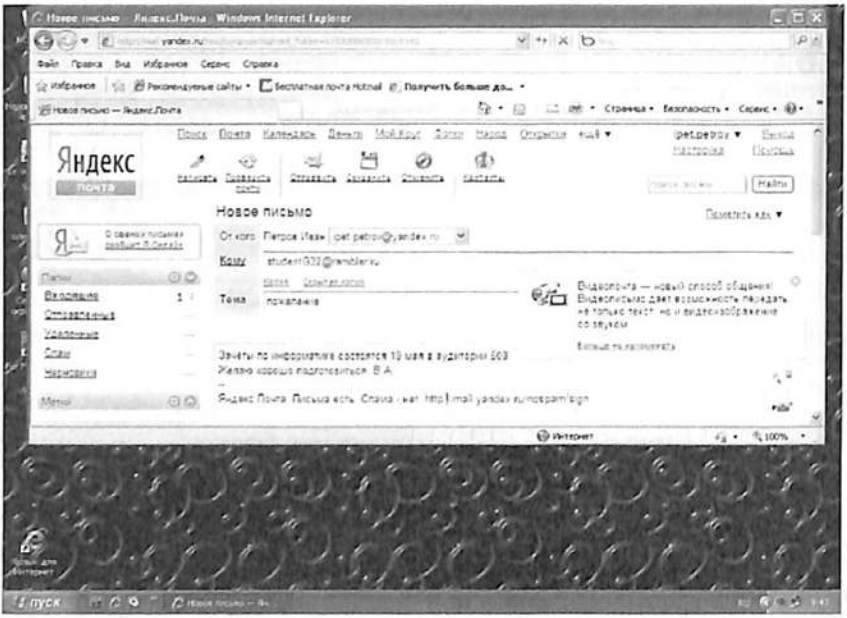


Рис. 7.6. Создание почтового сообщения в почте Яндекса

Контрольные вопросы

1. Каково назначение электронной почты?
2. Как создать логин и пароль?
3. Как создать сообщение для отправки в программе Outlook Express?
4. Как создать сообщение для отправки по почте системы Яндекс?

Часть III
АВТОМАТИЗИРОВАННЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ (АИС)

Глава 8
ОБЩАЯ ХАРАКТЕРИСТИКА АИС

8.1. Основные сведения об АИС, понятия и определения

Автоматизированные информационные системы являются, с одной стороны, разновидностью ИС (информационных систем — систем хранения, обработки и передачи информации), с другой — АС (автоматизированных систем — систем, в которых для выполнения технологических операций используются современные технические средства, заменяющие труд человека).

Автоматизация — это замена физического и умственного труда человека работой технических средств, обеспечивающих выполнение работ с заданной производительностью и качеством без вмешательства человека, за которым остаются функции наблюдения и подготовки технических средств к эксплуатации.

Слово «система» происходит от греч. *system* — целое, составленное из частей или множества элементов, связанных друг с другом и образующих определенную целостность, единство. Система — это совокупность взаимодействующих (взаимосвязанных) элементов, объединенных единством цели и общими целенаправленными правилами взаимоотношений.

Наличие взаимосвязей определяет организационную сложность системы. Локализация системы определяет границы систе-

мы, выделение ее элементов и связей (существенных и несущественных). Часто встречаются две ошибки: исключение существенных связей и учет несущественных связей. При построении системы должна быть определена целевая функция и разработаны алгоритмы структуры и функции системы. В ГОСТ 34.003—90 (приложение 1) дано следующее определение АИС.

«Автоматизированная система — система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций».

«Информационная технология — приемы, способы и методы применения средств вычислительной техники при выполнении функций сбора, хранения, обработки и использования данных» (ГОСТ 34.003—90). Следует четко знать разницу между информационной системой и информационной технологией.

При рассмотрении систем выделяют три основных научных направления.

Системный подход — основные задачи системного подхода состоят в разработке методов анализа и синтеза объектов, описания их целостных характеристик, исходя из целенаправленности поведения исследуемой системы и ее частей, имеющегося взаимодействия ее с окружающей средой.

Общая теория систем — основная задача общей теории систем состоит в том, чтобы, опираясь на понимание системы в виде комплекса взаимосвязанных элементов, найти совокупность законов, объясняющих поведение, развитие и функционирование системы.

Системный анализ — совокупность методов и методик выработки и принятия решений при проектировании, конструировании и управлении сложными объектами (социальными, экономическими, техническими и т. д.).

В вычислительной технике информационная система представляет собой программный комплекс, который предоставляет возможность надежного хранения данных в памяти, выполнения преобразований информации и вычислений, а также удобный и легко осваиваемый интерфейс. Таким образом, использование информационных систем позволяет:

- работать с большим объемом данных (причем прослеживается преобладание логической обработки над математической);
- хранить данные в течение длительного временного периода;

- связать несколько компонентов, имеющих свои локальные цели, задачи и приемы функционирования, в единую систему для работы с информацией;
- значительно снизить затраты на хранение и доступ к необходимым данным;
- достаточно быстро находить требуемую информацию и т. д.

Классические примеры информационных систем: банковские, фондового рынка, таможенной службы, статистические, резервирования авиационных или железнодорожных билетов и т. д.

Автоматизированная информационная система (АИС) — информационная система, использующая электронно-вычислительную машину на этапах ввода, подготовки и выдачи информации, т. е. она является развитием информационных систем, занимающихся поиском с помощью прикладных программ. Автоматизированные информационные системы относятся к классу сложных систем, как правило, в связи не столько с большой физической размерностью, сколько с многозначностью структурных отношений между их компонентами. Повышение эффективности функционирования предприятий невозможно без внедрения современных методов управления, базирующихся на АИС управления предприятиями. АИС позволяют:

- повышать производительность работы персонала, улучшать качество обслуживания клиентов;
- снижать трудоемкость и напряженность труда персонала, минимизировать ошибки в его действиях.

Таким образом, можно сделать вывод о том, что в общем виде автоматизированная информационная система — совокупность аппаратных (технических) средств, математических средств (математические модели, алгоритмические способы обработки информации, методы поиска и описания объектов программирования), телекоммуникационных средств, средств сбора и хранения информации.

Широко распространены АСУ, которым присущи многие функции АИС, но кроме них еще и функции управления различными объектами и процессами.

Таким образом, АИС — комплекс информационных, программных, технических, организационно-методических и других необходимых средств, обеспечивающих сбор, обработку, хранение, передачу данных, а также манипулирование ими для решения различных задач.

Управление — целенаправленное воздействие на любой самодвижущийся объект или процесс, в результате чего происходит как качественное, так и количественное изменение переменных, определяющих состояние объекта или процесса.

Выделяют два вида управления: предметами и людьми. В первом случае это управление орудиями производства и различными технологическими процессами. Во втором случае это управление группой людей (коллективом), обеспечивающее единство действий в целенаправленной работе.

Автоматизированная система управления (АСУ) — человеко-машинная система, реализующая автоматизированный сбор и переработку информации, необходимой для принятия решений по управлению объектом. АСУ создают для оптимального управления в различных сферах деятельности.

Автоматизированные информационные системы можно разделить на:

- системы информационного обеспечения, имеющие самостоятельное целевое назначение и область применения;
- системы (подсистемы) информационного обеспечения, входящие в состав автоматизированных систем управления (АСУ).

АИС первой группы, как правило, содержат информационную базу, используемую различными потребителями для удовлетворения информационных потребностей при принятии решений. Примером таких систем могут служить электронные библиотечные каталоги, АИС по законодательству («КонсультантПлюс», «Гарант» и др.), системы электронного документооборота финансовых документов («Система электронной обработки данных местного уровня» для автоматизации работы районных налоговых инспекций и др.).

К этой группе можно отнести следующие системы:

- информационно-справочные и информационно-поисковые;
- автоматизации документооборота;
- обучающие;
- экспертные;
- искусственного интеллекта;
- геоинформационные;
- гипертекстовые и др.

Информационно-справочные и информационно-поисковые системы (ИСС и ИПС) делят на документальные и фактографические.

Документальные системы — системы, предназначенные для поиска, обработки и вывода списков документов по определенным темам и признакам, полных текстов документов или их рефератов, справок различного назначения. Примером могут служить поисковые возможности системы «КонсультантПлюс».

Фактографические системы — системы, предназначенные для поиска, накопления, хранения, обработки и вывода данных по каким-либо фактам, событиям, сведениям или отдельному из них, выделенному из прочих.

Системы автоматизации документооборота — совокупность методов и средств для перевода документооборота из бумажной формы в электронную. Например, электронные депозитарии — базы данных, в которых хранятся записи об акционерах.

Обучающие системы — системы тренировочные и контролирующие, наставнические, имитационные и моделирующие, развивающие игры.

Тренировочные и контролирующие системы предназначены для закрепления умений и навыков на основе пройденного теоретического материала. Обучение идет за счет ответов обучаемых на предлагаемые вопросы. Если ответы неправильны, предлагаются подсказки.

Наставнические системы предназначены для изучения теоретического материала путем диалога человек—машина. Если ответы обучаемого неверны, программа предлагает повторно изучить материал.

Имитационные и моделирующие системы используют графически-иллюстративные и вычислительные возможности компьютерных программ и предназначены для построения моделей и ситуаций с возможностью изменения их параметров.

Развивающие игры предлагают обучаемому воображаемую среду, в которой он может участвовать, реализуя те или иные условия и комбинации.

Наиболее известные отечественные обучающие программы: «Урок», «Магистр», «Адонис» и др., зарубежные — Linkway, TeachCad и др. Многие из обучающих систем являются мультимедийными.

Экспертные системы (ЭС) — системы, которые с помощью ЭВМ и ПО выполняют функции экспертов при решении задач в области их компетенции. В экспертных системах накапливаются и могут долго храниться ценные данные и знания. В состав ЭС обычно входят база знаний и подсистемы вывода, объяснения,

приобретения знаний и др. Экспертные системы могут проводить анализ ситуации, выдавать советы и консультации, ставить объективный диагноз. Они решают задачи, которые решает специалист в результате проведения экспертизы. ЭС решают задачи на основе дедуктивных рассуждений с помощью эвристик (интуитивно найденных правил), поэтому могут находить решения задач, которые плохо определены и неструктурированны.

По степени автоматизации ЭС делят на:

- информационные — системы, включающие необходимую информацию для выработки решений, не затрагивая самой сути решений, которые после анализа принимает человек;
- информационно-советующие — системы, представляющие информацию для принятия решений и содержащие элементы оценки решений, но окончательное решение принимает человек;
- управляющие — системы, осуществляющие по заданным программам целенаправленное воздействие на производственный объект или процесс на основе исходной информации и выработанных решений;
- самонастраивающиеся — системы, которые могут в рамках заданного алгоритма изменить программу при ситуациях, не заданных в ней.

ЭС помогают организациям повышать квалификацию специалистов и эффективность работы. В настоящее время уже имеются тысячи экспертных систем, охватывающих самые разные предметные области. В качестве примеров можно привести Dendral — старейшая ЭС в области химии в мире, PROSPECTOR — система для коммерчески оправданного поиска полезных ископаемых, MYCIN — ЭС в области медицинской диагностики и многие другие.

Системы искусственного интеллекта — системы, в которых с помощью ЭВМ решаются сложные исследовательские задачи. Это задачи машинного перевода с одного естественного языка на другой, автоматического доказательства теорем, распознавания изображений, алгоритмы и стратегии игр, планирование действий роботов и др.

Искусственный интеллект — совокупность научных дисциплин, изучающих методы решения интеллектуальных (творческих) задач с использованием ЭВМ.

Геоинформационные системы — системы, в которых все данные об объектах привязаны к общей электронной топогра-

фической основе. Эти системы предназначены для использования в тех предметных областях, в которых структура объектов и процессов имеет пространственно-географическую привязку.

Гипертекстовые системы — системы с ассоциативным связыванием текстов, так называемым гипертекстом. Гипертекст — обычный текст, который содержит ссылки на связанные по смыслу фрагменты текста того же или другого документа. Гипертекстовые информационно-поисковые системы основаны на идее ассоциативно-навигационного подхода к анализу текстовой информации. Широкое применение они нашли в сети Интернет. С помощью текстового редактора (например, MultiEdit) или интернет-браузера пользователь, щелкнув мышью по выделенному цветом слову (по гиперссылке), может открыть связанный с этой ссылкой текст. Техника гипертекста стала в настоящее время основой для создания разных компьютерных справочных и учебных систем и энциклопедий.

АИС второй группы являются важнейшей составляющей различных АСУ:

- АСУП — АСУ предприятия;
- АСУТП — АСУ технологическими процессами;
- АСУТО — АСУ территориальными организациями и др.;
- ОГАС — общегосударственной автоматизированной системы;
- АСПР — автоматизированных систем плановых расчетов;
- АСГС — АС государственной статистики;
- САПР — систем автоматизированного проектирования;
- АСНИ — АС научных исследований и др.

В АСУ вычислительная техника используется в процессах не только сбора, хранения и обработки данных, но и принятия управленческих решений. АСУ базируются на использовании экономико-математических методов, средств вычислительной техники, средств получения и передачи данных. Особенностью является использование средств телекоммуникаций для получения данных с мест их возникновения, а также для отправки информации исполнителям и потребителям.

АСУ можно классифицировать по признакам назначения, ранга, характера действия, сложности и др.:

- по назначению: АСУ движущимися объектами, диспетчерские, организационные, предприятия, энергетическими установками, технологическими процессами и др.;

- по рангу (уровню управления): локальные (в рамках одной организации), региональные, отраслевые, межотраслевые, республиканские, общегосударственные и международные;
- по характеру действия: непрерывные и дискретные;
- по сложности: малые, средние, большие; и др.

В нашей стране действуют тысячи АСУ во всех отраслях экономики, культуры, образования, медицины.

Эффективно работает и совершенствуется, например, АСУ «Экспресс» — система обслуживания пассажиров и управления перевозками на железнодорожном транспорте. Эта АСУ представляет собой комплекс технических, программных, информационных, технологических и административных средств. Система базируется на ЭВМ единой серии, на единой международной нумерации пассажирских станций и на единой нумерации поездов. Система продажи билетов включает десятки тысяч касс и десять вычислительных центров (ВЦ). ВЦ имеют машинные вычислительные системы, устройства связи и коммутации (телеобработки). Билетные кассиры имеют возможность с помощью периферийной аппаратуры на своих автоматизированных рабочих местах (АРМ) выполнять различные операции по обслуживанию пассажиров.

АСУ «Сирена» — система обслуживания пассажиров Аэрофлота. Она предназначена для резервирования и учета мест на авиалайнерах, продажи билетов и выдачи информации о работе Аэрофлота в крупных городах. Система базируется на больших ЭВМ, взаимодействующих с большим количеством АРМ в пунктах продажи билетов на самолеты. Базы данных «Сирены» хранят годовое расписание авиарейсов, связывающих столицы СНГ и крупных городов России, данные о стоимости перевозок, данные о наличии свободных мест на каждый авиарейс и другую информацию. Обеспечивается актуализация баз данных.

Контрольные вопросы

1. Что такое автоматизация, система, управление?
2. Что понимается под совокупностью элементов, их взаимосвязанностью?
3. Что такое локализация системы и ее организационная сложность?
4. В чем заключается разница между информационной системой и информационной технологией?

5. Каково определение автоматизированной информационной системы?
6. На какие группы можно разделить автоматизированные информационные системы?
7. Какие системы можно отнести к каждой группе АИС?

8.2. Состав и структура АИС

Состав АИС:

- информационные ресурсы, представленные в виде баз данных (баз знаний), хранящих данные об объектах, связь между которыми задается определенными правилами;
- формальная логико-математическая система, реализованная в виде программных модулей, позволяющих осуществлять ввод, обработку, поиск и вывод необходимой информации;
- интерфейс, обеспечивающий общение пользователя с системой в удобной для него форме и позволяющий работать с информацией баз данных;
- персонал, определяющий порядок функционирования системы, планирующий порядок постановки задач и достижения целей;
- комплекс технических средств.

Информационные ресурсы включают машинную и немашинную информацию. Машинная информация представлена в виде баз данных, баз знаний, банков данных. Базы (банки) могут быть централизованные или распределенные.

Комплекс технических средств включает совокупность средств вычислительной техники (ЭВМ разных уровней, рабочие места операторов, каналы связи, запасные элементы и приборы) и специальный комплекс (средства получения информации о состоянии объекта управления, локальные средства регулирования, исполнительные устройства, датчики и устройства контроля и наладки технических средств).

Программное обеспечение состоит из общего ПО (операционные системы, комплексы программ технического обслуживания, специальные вычислительные программы) и специального ПО (организующие программы и программы, реализующие алгоритмы контроля и управления).

Персонал и инструктивно-методические материалы составляют организационное обеспечение системы.

Процедуры и технологии разрабатываются на основе логико-математических моделей и алгоритмов, составляющих основу математического обеспечения системы, и реализуются с помощью ПО и КТС, а также интерфейса, обеспечивающего доступ пользователя к информации.

Например, в состав экспертной системы (ЭС) входят:

- интерфейс, позволяющий передавать в базу данных информацию и обращаться к системе с вопросом или за объяснением;
- рабочая память (БД), которая хранит данные об объектах;
- диспетчер, определяющий порядок функционирования ЭС;
- машина вывода — формально-логическая система, реализованная в виде программного модуля;
- база знаний (БЗ) — совокупность всех имеющихся сведений о предметной области, записанных с помощью формальных структур представления знаний (набора правил, фреймов, семантических сетей и др.).

Важнейшей составляющей ЭС является блок объяснений. Он позволяет пользователю задавать вопросы и получать разумные ответы.

Структура АИС — определенное устройство системы изнутри.

АИС состоит, как правило, из функциональной и обеспечивающей частей, каждая из которых имеет свою структуру.

Функция есть проявление взаимодействия системы с внешней средой. Проявление функции во времени называется функционированием.

Функциональная часть — совокупность подсистем, зависящих от особенностей АСУ. Эти подсистемы разделяются по определенному признаку (функциональному или структурному) и объединяют в себе соответствующие комплексы задач управления.

Обеспечивающая часть — совокупность информационного, математического, программного, технического, правового, организационного, методического, эргономического, метрологического обеспечения.

Структура АИС представлена на рис. 8.1.

Обеспечивающая часть. Информационное обеспечение АИС — это совокупность баз данных и файлов операционной системы, форматной и лексической баз, а также языковых средств, предназначенных для ввода, обработки, поиска и представления информации в форме, необходимой потребителю. Подробно об информационном обеспечении (ИО) см. в разделе 9.1.

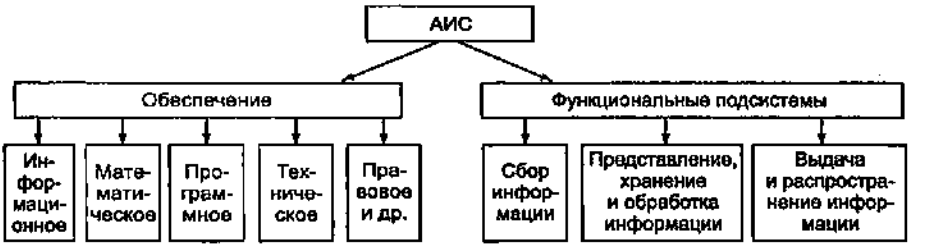


Рис. 8.1. Структура АИС

ИО включает массивы форматированных (и неформатированных) документов, классификаторы, кодификаторы, словари, нормативную базу для реализации решений по объемам, размещению и формам существования информации в АИС, а также совокупность средств и правил для формализации естественного языка, используемых при общении пользователей и эксплуатационного персонала АС с комплексом средств автоматизации.

В настоящее время ИО рассматривают как совокупность собственно ИО и лингвистического обеспечения. При этом собственное ИО включает файлы операционных систем и БД, а лингвистическое — форматную базу, лексическую базу и языковые средства.

Математическое обеспечение — «совокупность математических методов, моделей и алгоритмов, примененных в АС» (ГОСТ 34.03—90).

Программное обеспечение — совокупность общесистемных и прикладных программ, а также инструктивно-методической документации по их применению.

Техническое обеспечение — комплекс технических средств, обеспечивающих работу системы. Это технические средства сбора, регистрации, передачи, обработки, отображения, размножения информации.

Правовое обеспечение — совокупность нормативно-правовых документов, определяющих права и обязанности персонала в условиях функционирования системы, а также комплекс документов, регламентирующих порядок хранения и защиты информации, правил ревизии данных, обеспечение юридической чистоты совершаемых операций.

Организационно-методическое обеспечение — совокупность документов, определяющих организационную структуру системы автоматизации для выполнения конкретно автоматизируемых функций.

Эргономическое обеспечение — совокупность методов и средств по созданию оптимальных условий для работы специалистов в рамках АИС.

Метрологическое обеспечение — методы и средства метрологии и инструкции по их применению для всех компонентов АИС.

Функциональная часть. Функция системы — совокупность действий, направленных на достижение определенной частной цели. Функции АИС подразделяются на информационные, управляющие, защитные и вспомогательные.

Информационные функции реализуют сбор, обработку и представление информации о состоянии автоматизируемого объекта оперативному персоналу или передачу этой информации для последующей обработки. Это могут быть следующие функции: измерение параметров, контроль, вычисление параметров, формирование и выдача данных оперативному персоналу или в смежные системы, оценка и прогноз состояния АС и ее элементов.

Управляющие функции вырабатывают и реализуют управляющие воздействия на объект управления. Это могут быть регулирование параметров, логическое воздействие, программное логическое управление, управление режимами, адаптивное управление.

Защитные функции могут быть технологические и аварийные.

При автоматизированной реализации функций различают следующие режимы:

- диалоговый (персонал имеет возможность влиять на выработку рекомендаций по управлению объектом с помощью ПО и КТС);
- советчика (персонал принимает решение об использовании рекомендаций, выданных системой);
- ручной (персонал принимает управляющие решения на основе контрольно-измерительной информации).

Подсистемы функциональной части системы строят в соответствии с информационными и управляющими функциями.

Подсистема сбора информации осуществляет сбор информации по каналам связи разными способами: ручным, автоматизированным, иногда автоматическим.

Операторы выполняют первичный сбор и систематизацию информации. Собранная информация анализируется с точки зрения выявления сущностей, которые будут являться прообра-

зами создаваемых таблиц БД (если БД реляционная). Далее информация направляется в подсистему представления, хранения и обработки информации.

Подсистема представления, хранения и обработки информации выполняет предмашинную подготовку данных и ввод их в базу данных, рассматриваемую как информационная модель предметной области. Операторы при участии администратора базы данных по определенным правилам на основе инструкций заполняют базу данных подготовленной информацией. В этой подсистеме осуществляется проверка данных на достоверность и непротиворечивость, редактирование, обработка и анализ данных, сохранность накапливаемых данных, восстановление утерянных. Основой этой подсистемы является информационный фонд — база данных (БД), управляющая СУБД (см. гл. 6).

В совокупности информационный фонд, а также средства и методы его обработки могут представлять собой банк данных.

Характер представления информации и ее логическая организация в БД зависят от типа системы.

В документальных АИС документы при вводе в БД не структурируются. Для поиска документа создается его поисковый образ (ПОД) — карточка реквизитов.

В фактографических АИС данные при вводе в БД структурируются и хранятся в виде множества экземпляров структурных элементов, которые содержат сведения по фактам, событиям. Структура объекта включает конечный набор реквизитов (характеристик). Например, хранятся данные о персонале. Запись о каждом сотруднике содержит такие реквизиты, как фамилия, имя, отчество и т. д.

В геоинформационных системах данные структурированы в отдельные информационные объекты, которые привязаны к общей топографической основе (электронной карте).

В этой подсистеме особое внимание уделяется вопросам поддержания целостности БД, т. е. достоверности и непротиворечивости хранимой и обрабатываемой информации, а также вопросам защиты информации.

Подсистема выдачи и распространения информации осуществляет поиск необходимых данных по запросам, создание готовых документов и отчетов, передает готовые документы по каналам связи и предоставляет требуемую информацию потребителям.

Приведенная выше схема структуры АИС осуществляется в основном в информационно-справочных, информационно-по-

исковых системах. Структура более сложных систем, по существу, представляет собой АИСУ, т. е. АИС управления, АСУ различных уровней и назначения. Например, АИСУ «Налог» представляет собой систему организационного управления органами Госналогслужбы. Это многоуровневая система, осуществляющая:

- на первых (высших) уровнях (Президент РФ, Правительство РФ, Государственная налоговая служба РФ) — методологическое руководство и контроль за налогообложением по разным видам налогов на уровне страны;
- второй уровень (налоговые службы краев и областей, налоговые службы республик, налоговые службы Москвы и Санкт-Петербурга) — методическое руководство и контроль над налогообложением по разным видам налогов на уровне территорий;
- третий уровень (налоговые инспекции районов, налоговые инспекции городов, налоговые инспекции городских районов) — непосредственное взаимодействие с налогоплательщиками.

В налоговой системе процесс управления является информационным. АИС налоговой службы состоит из обеспечивающей и функциональной частей.

Обеспечивающая часть включает информационное, программное, техническое и другие виды обеспечения, характерные для АИС организационного типа.

Функциональная часть отражает предметную область и представляет собой совокупность подсистем, зависящих от особенностей системы. Каждому уровню АИСУ соответствует свой состав функционального обеспечения.

Так, на втором уровне структура системы выглядит следующим образом (рис. 8.2).

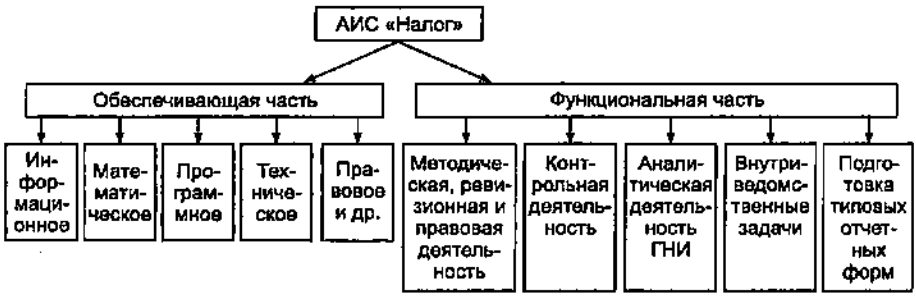


Рис. 8.2. Структура АИСУ «Налог» (второй уровень)

Подсистема методической, ревизионной и правовой деятельности обеспечивает работу с законодательными актами, постановлениями, указами и другими правительственными документами, а также с нормативными и методическими документами Госналогслужбы РФ. В подсистеме проводят сбор, обработку и анализ информации, поступающей от территориальных налоговых инспекций.

Подсистема контрольной деятельности обеспечивает документальную проверку предприятий и ведение Государственного реестра предприятий и физических лиц. Реестр предприятий содержит официальную регистрационную информацию о предприятиях (юридических лицах), а реестр физических лиц — информацию о налогоплательщиках, обязанных представлять декларацию о доходах и уплачивать отдельные виды налогов с физических лиц.

Подсистема аналитической деятельности Государственных налоговых инспекций (ГНИ) предусматривает анализ динамики налоговых платежей, прогнозирование величины сбора отдельных видов налогов, экономический и статистический анализ хозяйственной деятельности предприятий региона, определение предприятий, подлежащих документальной проверке, анализ налогового законодательства и выработку рекомендаций по его совершенствованию, анализ деятельности территориальных налоговых инспекций и др.

Подсистема внутриведомственных задач решает задачи, обеспечивающие деятельность аппарата ГНИ, и включает в себя делопроизводство, бухгалтерский учет, материально-техническое снабжение, работу с кадрами и др.

Подсистема подготовки типовых отчетных форм формирует сводные таблицы статистических показателей, которые характеризуют типовые виды деятельности ГНИ регионального уровня по сбору различных видов налоговых платежей, и контролирует этот процесс.

Структура системы на третьем уровне включает другие функциональные подсистемы:

- регистрации предприятий;
- камеральной проверки;
- ведения лицевых карточек предприятий;
- анализа состояния предприятия;
- документальной проверки;
- ведения нормативно-правовой документации;

- внутриведомственных задач;
- обработки документов физических лиц.

Подробно описывать эти подсистемы здесь не представляется целесообразным.

Отметим, что функциональные подсистемы состоят из комплексов задач, которые характеризуются определенным экономическим содержанием и достижением конкретной цели. В комплексе задач используются различные первичные документы и составляются выходные документы на основе взаимосвязанных алгоритмов расчетов, которые базируются на методических материалах, нормативных документах, инструкциях и т. п.

Рассматривая АИС как информационную автоматизированную систему управления предприятием (АСУП), можно, например, представить ее структуру в виде, изображенном на рис. 8.3.

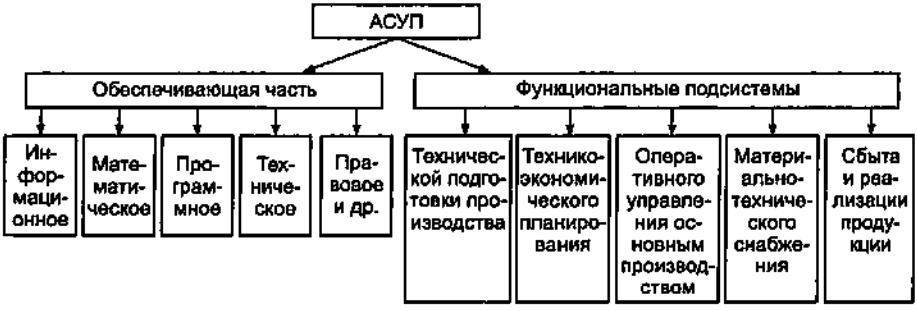


Рис. 8.3. Структура АСУП

Естественно, могут быть и другие функциональные подсистемы.

АСУ, как и любую систему управления, удобно рассматривать как некоторую совокупность процессов и объектов (взаимосвязанных элементов). Каждая из подсистем является обособленной и может рассматриваться как часть (подсистема) системы более высокого уровня.

АСУ строится по иерархическому принципу (многоуровневого подчинения) взаимосвязи как по структурному местоположению, так и по распределению функций управления. Систему можно представить как композицию подсистем различных уровней. Для получения элементарных составляющих системы выполняют ее декомпозицию, образуя дерево метасистемы, на котором выделяются подсистемы различных уровней.

Декомпозиция осуществляется по функциям или составу элементов (данные, информация, документы, технические средства, организационные подразделения и т. д.).

Промышленные предприятия — сложная производственная система управления, включающая следующие процессы: информационные, технические, организационные, экономические. Для проектирования АСУ создают модель производственной системы, первоначально исходя из ее структуры, учитывая тесную связь логических и физических уровней. Используют связь типа «один ко многим».

В функциональных подсистемах АСУ выполняются работы по следующим направлениям:

1. Планирование производства и выпуск продукции.
2. Обеспечение производства основными средствами, трудовыми ресурсами, оборотными фондами.
3. Планирование себестоимости продукции, финансов, сбыта продукции.

Например, АСУ «Прибор» — отраслевая система управления деятельностью отрасли приборостроения в целом и отдельными предприятиями.

В АСУ «Прибор» выполнялось оптимизированное решение задач разных подсистем. Система стала базовой для разработки АСУ в других отраслях хозяйства.

В основу разработки системы положен организационно-функциональный принцип, отражающий структуру Министерства приборостроения, средств автоматизации и систем управления. Например, наряду с другими подсистемами для решения задач технико-экономического планирования (ТЭП) были созданы следующие подсистемы:

- перспективного планирования и размещения отрасли;
- технико-экономического планирования и анализа показателей плана;
- планирования, учета и анализа труда и заработной платы;
- планирования и учета кадров.

Первая очередь была введена в эксплуатацию в 1970 г. и решала 74 задачи, связанные с учетом, планированием и управлением производством.

Вторая очередь задействована в 1975 г. Решались уже 292 задачи. Руководители получили возможность через выносные экраны получать в запросно-ответном режиме необходимые для принятия решений данные.

Главный вычислительный центр АСУ «Прибор» обслуживал руководство министерства по 17 функциям управления и 12 всесоюзных промышленных объединений. На ЭВМ выполнялись оптимизационные расчеты перспективных и годовых планов предприятий, подотраслей и отраслей в целом с определением темпов роста производства, основных пропорций и направлений развития отдельных отраслей. По каждому предприятию рассчитывалось до 20 вариантов перспективных и 5—6 вариантов годовых планов. Оптимизация таких важных показателей, как объем реализуемой продукции, прибыль от реализации и др., позволили принимать более рациональные решения. Рассчитанные варианты производственных планов технико-экономических показателей улучшили распределение финансов и трудовых ресурсов.

Контрольные вопросы

1. Что входит в состав АИС?
2. Каковы информационные ресурсы АИС?
3. Что включает комплекс технических средств АИС?
4. На основе чего разрабатываются процедуры и технологии АИС?
5. Какова типовая структура АИС?
6. Что включает обеспечивающая часть АИС? Охарактеризуйте ее.
7. Что включает функциональная часть АИС? Охарактеризуйте ее на примере системы.

8.3. Методы, стадии и этапы создания АИС

Разработка и проектирование АИС начинается с создания концептуальной модели использования системы. Прежде всего, должны быть определены целесообразность создания системы, ее конкретные функции и подлежащие автоматизации задачи. Должна быть выполнена оценка не только целей, но и возможностей создания системы. Далее проводится анализ требований к АИС, детальное проектирование, взаимосвязь этапов, программирование и тестирование, минимизация потерь при переходе от одного уровня представления информации к другому, интеграция в существующую систему, внедрение и поддержка.

Существует три класса методологий проектирования АИС:

- концептуальное моделирование предметной области;
- выявление требований и спецификация информационной системы через ее макетирование;
- системная архитектура программных средств, поддерживаемая инструментальными средствами CASE-технологии (CASE — Computer Aided Software Engineering — технология создания и сопровождения ПО различных систем).

Современные методологии проектирования систем должны обеспечивать описание объектов автоматизации, описание функциональных возможностей АИС, спецификацию проекта, гарантирующую достижение заданных характеристик системы, детальный план создания системы с оценкой сроков разработки, описание реализации конкретной системы.

Спецификация — точное, полное, ясно сформулированное описание требований для данной задачи.

Существует три *метода разработки АИС*: оригинальный, типовой, автоматизированный.

Метод оригинального проектирования охватывает все виды работ для различных объектов, выполняемых по специальным проектам, включающим оригинальные методики и средства выполнения работ. Методики на всех этапах работ создаются для конкретного объекта по мере необходимости. Недостатками этого метода являются высокая трудоемкость, большие сроки проектирования, плохие модернизируемость и сопровождаемость.

Метод типового проектирования предполагает разбиение системы на отдельные модули (элементы, подсистемы, объекты) и разработку для каждого из них законченного проекта. Это позволяет при внедрении адаптировать каждый модуль к конкретным условиям функционирования системы. Например, элементами могут быть ИО, ПО, ТО и др.

Подсистемами могут выступать функциональные подсистемы сбора информации, распространения информации и т. д.

Метод автоматизированного проектирования предполагает автоматизацию основных этапов создания АИС, начиная от выбора состава задач и заканчивая автоматическим получением проектной документации. Для реализации этого метода используют представленные на ЭВМ типовые проекты и типовые проектные решения, ППП, ОС, САПР, CASE-технологии.

Процесс создания АИС многообразен и по времени довольно продолжителен. Он требует достаточно больших трудовых и

денежных затрат. Этот процесс делят на стадии и этапы, на каждом из которых в соответствии с поставленными целями и решаемыми задачами работают специалисты разного профиля и уровня.

Стадия создания автоматизированной системы — часть процесса создания АС, установленная нормативными документами и заканчивающаяся выпуском документации, которая должна содержать модель системы на уровне данной стадии, изготовление несерийных компонентов, приемку системы в эксплуатацию.

Каждая стадия выделена по соображениям рационального планирования и организации работ и обязательно должна заканчиваться определенным результатом. Содержание документации на каждой стадии определяется составом и спецификой работ.

В ГОСТ 34.601—90 определено 8 стадий создания автоматизированных систем:

1. Формирование требований к АС.
2. Разработка концепции АС.
3. Техническое задание.
4. Эскизный проект.
5. Технический проект.
6. Рабочая документация.
7. Ввод в действие.
8. Сопровождение АС.

Можно выделить три периода создания системы: предпроектный, проектирование, ввод в эксплуатацию.

Стадии 1—3 относятся к первому периоду, стадии 4—6 — ко второму периоду, стадии 7, 8 — к третьему.

В предпроектный период разрабатывают технико-экономическое обоснование (ТЭО) и ТЗ на проектирование системы.

В этот период на стадии формирования требований к АС проводят три этапа работ:

- обследование объекта предметной области и обоснование необходимости создания системы;
- формирование требований пользователей к системе;
- составление отчета о выполненной работе и заявки на разработку системы.

На стадии разработки концепции АС проводят четыре этапа работ:

- изучение объекта;
- проведение научно-исследовательских работ;

- выбор варианта концепции системы из нескольких разработанных;
- составление отчета о выполненной работе.

На стадии 3 разрабатывают и утверждают техническое задание на создание АС.

Техническое задание (ТЗ) — это перечень основных эксплуатационных, технологических, экономических и других требований, которым должен удовлетворять проектируемый объект на всех этапах его существования.

После утверждения ТЗ начинается второй период создания АС — период проектирования системы.

Проектирование — процесс обоснованного выбора характеристик системы, формирования логико-математических и экономико-математических моделей, разработки документации.

В начале проектирования разрабатывают документацию, достаточную для утверждения намеченных затрат, доходов, дополнительной численности персонала, дополнительных площадей и организационно-технических решений, а при дальнейшем проектировании — для заказа и комплектации оборудования и материалов, его монтажа и наладки, для организации работы АСУ и ее внедрения.

Проектирование АС должна выполнять специализированная организация-проектировщик АС, соисполнителем может быть генпроектировщик отрасли.

На стадии создания эскизного проекта на первом этапе разрабатывают предварительные проектные решения по системе и ее частям, на втором — документацию на АС и ее части.

На стадии 5 при создании технического проекта в четыре этапа проводят разработку:

- проектных решений по системе и ее частям;
- документации на АС и ее части;
- документации на поставку изделий для комплектования АС и ТЗ на их разработку;
- заданий на проектирование в смежных частях проекта объекта автоматизации.

Технический и рабочий проекты должны пройти экспертизу и утверждение, т. е. оценку экономической эффективности проекта специалистами высокого ранга и класса различных организаций с целью сведения к минимуму всех видов потерь. После утверждения проекта предприятие получает средства для финансирования.

Экспертизу и утверждение проекта выполняют генпроектировщик, автоматизируемое предприятие и специализированная проектная организация.

На стадии 6 выполняется разработка рабочей документации. На первом этапе создают рабочую документацию на систему и ее части. На втором этапе осуществляют разработку или адаптацию программ.

Третий период — ввод в эксплуатацию АС. Обеспечивают разработку нестандартного оборудования, комплектацию оборудования, материалов, покупных изделий, монтаж, наладку, внедрение.

Разработка нестандартного оборудования — задание и выполнение НИР (научно-исследовательских работ) и ОКР (опытно-конструкторских работ) на необходимое нестандартное оборудование, включая программное обеспечение.

Разработку нестандартного оборудования выполняют СКБ (специальные конструкторские бюро), ОКБ и НИИ Министерства приборостроения, средств автоматизации и систем управления; комплектацию — предприятие через государственные и частные фирмы.

Монтаж начинается после изготовления нестандартного оборудования и комплектации стандартного. Проводит его специализированная монтажная организация.

Наладка — приведение в рабочее состояние всех технических средств и обеспечение готовности их работы в процессе опробования испытаний и принадлежностей.

Внедрение — процесс перехода к практическому применению решений проекта, когда технические средства системы используются для управления производством по новой технологии, разработанной в проекте.

На стадии 7 система вводится в эксплуатацию в восемь этапов:

- подготовка объекта автоматизации к вводу АС;
- подготовка персонала;
- комплектация АС программными, техническими, информационными средствами и изделиями;
- строительно-монтажные работы;
- пусконаладочные работы;
- предварительные испытания;
- опытная эксплуатация;
- приемочные испытания.

На стадии 8 предусматриваются работы на первом этапе по обязательствам гарантийного обслуживания и на втором — по обязательствам послегарантийного обслуживания. Осуществляется авторский надзор генеральным проектировщиком АС (специализированная проектная организация), генеральным проектировщиком отрасли и разработчиком нестандартного оборудования.

С целью улучшения управления ходом проектирования каждая стадия детализируется, т. е. разбивается на этапы.

Этап создания автоматизированной системы — часть стадии создания АС, определяемая по характеру работ, его результату или специализации исполнителей.

Содержание этапов создания АС регламентировано в Приложении 1 ГОСТ 34.601—90.

Приведенные стадии и этапы при разработке систем могут не всегда быть реализованы полностью. Разработка проходит только все необходимые для конкретной системы стадии и этапы.

Например, рассматривая этапы разработки экспертных систем, выделяют следующие:

1. Идентификация — определение проблемы и ее концептуальное описание, определение ресурсов, целей, экспертов.

2. Концептуализация — выделение ключевых понятий системы, отношений и характеристик, достаточных для полного и детального описания объекта.

3. Формализация и моделирование — описание введенных понятий на некотором формальном языке, построение модели исследуемой области.

4. Рабочее проектирование — разработка документации для создания общего и функционального обеспечения системы.

5. Выполнение — создание одного или нескольких прототипов системы.

6. Тестирование — оценка выбранного метода представления знаний и работоспособности всей системы в целом на основе проверки прототипа.

7. Опытная эксплуатация — проверка пригодности системы для конечного пользователя.

8. Модификация системы — введение изменений в модель и документацию системы, полное или частичное перепрограммирование и доведение прототипа до состояния нового программного продукта.

Стандартизация и типизация, использование типовых проектных решений как составных частей или блоков при проектировании АС для конкретных проектов позволяют значительно сократить трудоемкость разработки и время внедрения, получить большее разнообразие проектных решений.

Создание АС предусматривало разработку комплексов проектных решений трех классов:

1. Комплекс задач и методик обработки данных, алгоритмов и программ для информационного обслуживания и реализацию функций управления. Проектные решения могут быть здесь развиты до масштабов подсистем или информационных систем.

2. Определение структуры, состава, размещения и порядка использования ТС и ПО, которые предназначены для предварительной подготовки данных, их передачи между компонентами системы для решения задач обработки данных и выдачи ее результатов потребителю.

3. Определение организационных решений и действий персонала на всех стадиях разработки, внедрения и функционирования отдельных задач, прикладных систем и системы в целом.

Контрольные вопросы

1. Каковы классы методологий проектирования АИС?
2. Каковы методы разработки АИС? Дайте их характеристику.
3. Что такое стадия и этап создания автоматизированной системы?
4. Сколько существует стадий создания АИС? Приведите перечень и краткую характеристику.
5. Что такое техническое задание на создание АС и из каких разделов оно состоит?
6. Что такое проектирование системы?
7. Какие комплексы проектных решений предусматривает создание АИС?

Глава 9
ОБЕСПЕЧИВАЮЩАЯ И ФУНКЦИОНАЛЬНАЯ ЧАСТИ АИС

9.1. Информационное обеспечение

Основной принцип создания ИО — решение задачи удовлетворения информационных потребностей пользователя и/или системы управления объектом (производством).

В ИО выполняются следующие процессы:

- накопление информации;
- обмен информацией;
- обработка информации;
- управление данными;
- формализация данных и знаний.

Создание ИО проходит следующие этапы:

- исследование информационных потоков;
- разработка системы классификации и кодирования;
- разработка унифицированных форм представления данных в информационной базе;
- накопление массивов данных и работа с ними.

На предпроектной стадии разработки АС на основании технико-экономического обоснования разрабатывается техническое задание (ТЗ) на создание системы. В ТЗ определяются принципы построения системы, организационная и функциональная структуры, требования к обеспечивающим подсистемам, в том числе к ИО.

Информационное обеспечение определяет размещение и формы организации информации в системе, представляя собой совокупность методов и средств построения и реализации информационной базы. ИО реализуется как внешнее (немашинное) и внутримашинное.

1. Внешнее (немашинное) ИО, которое, однако, должно учитывать принципы автоматизации информационных процессов. Его состав:

- СКК (система классификации и кодирования);
- НСД (нормативно-справочные документы);
- ОД (оперативные документы);
- ММ (методические и инструктивные материалы).

Движение этих документов реализуется в соответствии с организационной структурой управления.

2. Внутримашинное ИО включает:

- ИМ (информационные массивы), составляющие информационную базу системы;
- ПП (пакеты программ).

Информационное обеспечение реализуется в виде банков данных и банков знаний, в основе построения которых — модели накопления данных и представления знаний. Эти процессы должны быть формализованы на концептуальном и логическом уровнях.

Традиционно, в соответствии с ГОСТ 34.03—90 информационное обеспечение рассматривалось как «совокупность форм документов, классификаторов, нормативной базы и реализованных решений по объемам, размещению и формам существования информации, применяемой в АС при ее функционировании».

ИО АИС включало три составляющие:

- единую систему классификации и кодирования информации;
- унифицированные системы документации;
- массивы информации.

Лингвистическое обеспечение рассматривалось как «совокупность средств и правил для формализации естественного языка, используемых при общении пользователей и эксплуатационного персонала АС с комплексом средств автоматизации при функционировании АС». Лингвистическое обеспечение АИС включало две составляющие: лексическую (словарную) базу и языковые средства.

В настоящее время ИО рассматривается как совокупность собственно информационного обеспечения и лингвистического обеспечения. При этом под собственно информационным обеспечением понимают файлы операционной системы и базы данных, а под лингвистическим — форматную базу, лексическую базу и языковые средства.

Поэтому можно сформулировать следующее утверждение.

Информационное обеспечение АИС — это совокупность баз данных и файлов операционной системы, форматной и лексической баз, а также языковых средств, предназначенных для ввода, обработки, поиска и представления информации в форме, необходимой потребителю.

Данные, отображающие сведения об определенной предметной области, могут поступать из внешней и внутренней среды системы как в неструктурированном виде (например, различные документы на естественном языке и др.), так и в структурированной форме (анкеты, таблицы и др.). Естественно, что способы сбора и обработки таких данных отличаются друг от друга. Эти данные необходимо так структурировать, т. е. создать такие структурированные документы, чтобы стала возможной их программная обработка. В общем случае документ представляет собой зафиксированную на материальном носителе информацию (данные) с реквизитами, позволяющими ее идентифицировать.

Сбор информации осуществляется от источников по каналам получения информации. Отнесение информации к определенной предметной области — сложная классификационная и плохо поддающаяся автоматизации задача, поэтому данная операция, как правило, выполняется специалистами.

Затем осуществляется комплектование БД, т. е. выполняется предварительная обработка и рубрикация информации. Далее неструктурированная информация подлежит структуризации.

Структуризация информации — процесс представления неформализованной документированной информации на информационном языке представления данных в конкретной АИС.

Структурированная информация заносится в БД системы, и устанавливается ее связь с уже имеющейся в базе информацией.

Структура БД организована в зависимости от типа модели данных (иерархическая, сетевая, реляционная). Определение каждой модели, а также основные сведения о БД приведены в главе 6. Пример иерархической модели представлен на рис. 9.1.

В иерархической модели соблюдается строгая последовательность обхода по вертикали или горизонтали. Операции над данными имеют строгую определенность: найти указанное дерево, в этом дереве найти указанный уровень, в уровне найти указанную запись и т. д. Соблюдается строгая последовательность перехода от родительской сущности к дочерней.

Рис. 9.1. Иерархическая модель организации для определения структуры БД

Но рассматривается и общая для БД логическая структура. БД включает одну или несколько подбаз (файлов, таблиц, массивов). Каждая подбаза состоит из агрегатов данных (записей, документов). Запись состоит из полей. Поля могут быть элементарные (имеют фиксированную и ограниченную длину), составные (агрегаты элементарных), текстовые (имеют переменную длину и сложную внутреннюю структуру), бинарные (данные, рассматриваемые как поля).

К физической структуре БД относят файлы первичных (исходных) данных, файлы вторичной (справочной) информации, тезаурусы и словари данных (см. описание лексической базы), индексы.

Файлы исходных данных содержат объекты, подлежащие обработке. Файлы вторичной информации содержат описания объектов или их элементов.

Индекс — указатель (файл), связывающий адрес объекта с его содержанием. Включает список и частотный словарь.

В настоящее время для создания баз данных (БД) АИС используют различные СУБД.

В главе 6 рассмотрена методика создания реляционной БД под управлением СУБД Access. Рассмотрим методику создания реляционной БД в среде СУБД FoxPro.

Прежде всего проектируют реляционную структуру БД и выполняют нормализацию таблиц.

Первый этап проектирования БД — построение концептуальной информационной модели организации. Для этого должны быть изучены концептуальные требования заказчика (организации) и на основе анализа этих требований определены сущности. Результатом работ первого этапа проектирования БД должен

быть список основных сущностей — прообраз будущих таблиц и информационная (концептуальная) модель данных.

Второй этап проектирования — определение взаимосвязей между сущностями. Результатом работ второго этапа проектирования БД должна быть схема, отражающая взаимосвязи между сущностями.

Третий этап проектирования — задание первичных и внешних ключей для перехода между сущностями. Результатом работ третьего этапа проектирования БД должна быть общая таблица с описанием всех сущностей — прообраз будущих таблиц. В таблице кроме атрибутов (будущих полей) задаются первичные и внешние ключи для каждой таблицы.

Четвертый этап проектирования — приведение модели к требуемому уровню нормальной формы, т. е. выполнение нормализации отношений между таблицами. Следует удалить из БД избыточную информацию. Для этого нужно создать для каждой сущности по одной таблице с тем же именем, а полями будут атрибуты сущности. При этом следует выполнить условия:

1) первой нормальной формы таблицы:
   - каждое поле должно быть неделимо;
   - не должно быть повторяющихся полей или групп полей;
2) второй нормальной формы таблицы:
   - все условия первой нормальной формы;
   - первичный ключ однозначно определяет всю запись;
   - все поля зависят от первичного ключа;
   - первичный ключ не должен быть избыточен;
3) третьей нормальной формы таблицы:
   - все условия второй нормальной формы;
   - каждое неключевое поле не должно зависеть от другого неключевого поля.

Пятый этап проектирования — описание каждой таблицы: присвоение имен таблицам и полям, определение типа и размера полей, указание полей, по которым надо построить ключи и индексы, определение виртуальных полей, указание назначения каждого поля. Результатом работ пятого этапа проектирования БД должны быть нормализованные таблицы с полным описанием всех их элементов.

После проектирования БД выполняют ее создание.

Пример созданных таблиц приведен на рис. 9.2.

Создать файл БД, открыть таблицы и работать с записями можно двумя способами: с помощью специальных команд и с помощью главного меню.

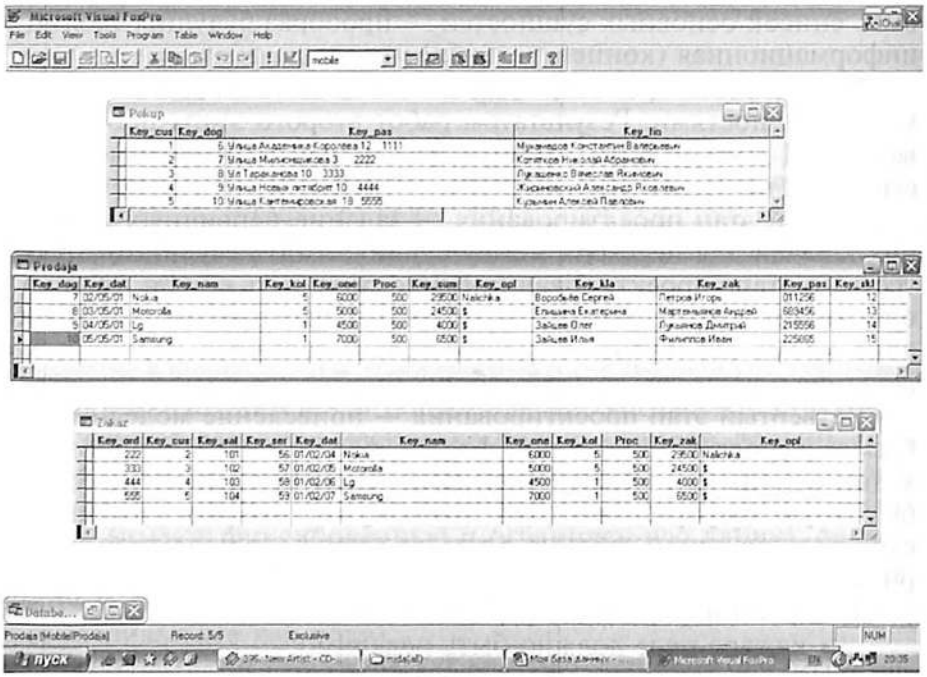


Рис. 9.2. Вид созданных таблиц

Для облегчения поиска данных в таблице выполняют индексирование таблиц.

Индексы создаются по значениям одного поля (простой) или нескольких полей (сложный). Во время построения индекса записи в таблице сортируются по значениям поля (или полей) будущего индекса. Индекс (ключ) имеет свой тип (`Type`): первичный (`Primary`) — только один, уникальный; внешние — `Candidate`, `Unique` или `Regular`. Если построен один индекс, то он хранится в одноиндексном файле, имеющем расширение .idx. Файлы, хранящие много индексов, называются мультииндексными и имеют расширение .cdx. Создать индекс можно с помощью командной строки или главного меню (рис. 9.3, 9.4).

Сортировку данных в таблицах осуществляют по возрастанию или убыванию двумя способами: в соответствии с индексом или с помощью команды `SORT`.

При поиске данных в таблицах используют два метода:

- последовательного (полного) перебора;
- деления пополам (по полю текущего индекса).

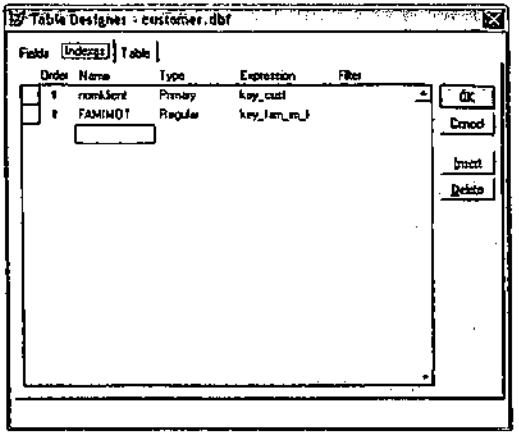


Рис. 9.3. Создание индексов с помощью диалогового окна **Table Designer (Конструктор таблиц)**

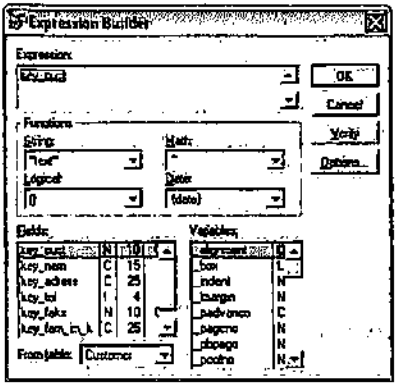


Рис. 9.4. Создание сложных индексов с помощью диалогового окна **Expression Builder (Построитель выражений)**

Поиск методом полного перебора производится по любому полю таблицы с помощью определенных команд или при выборе из главного меню команд **Table → Go to Record → Locate (Таблица → Перейти к записи → Найти)**.

Поиск данных в таблицах по полю текущего индекса (метод деления пополам) выполняется также с помощью определенных команд или при выборе из главного меню команд **Edit → Find (Правка → Найти)**.

Фильтрация данных осуществляется с помощью фильтров двух видов:

- фильтр для строк, когда ограничивается количество строк;
- фильтр для полей, когда ограничивается количество полей, отображаемых на экране.

Для установки фильтра данных используют команду `SET FILTER TO <выр. L>`.

Очень важным моментом является установление взаимосвязей между таблицами.

Для одновременной работы с несколькими таблицами нужно поместить каждую таблицу в свою рабочую область и установить взаимосвязи между ними. Указатели записей во взаимосвязанных таблицах будут двигаться синхронно.

В старшей таблице указатель перемещается произвольно. В младшей или подчиненной таблице указатель перемещается в

соответствии с перемещением указателя в старшей таблице. К одной старшей таблице можно подключать несколько младших.

Родительская таблица должна иметь первичный ключ (индекс). Дочерняя таблица должна иметь внешний ключ (индекс). Одна запись в родительской таблице порождает несколько записей в дочерней. Общее поле, т. е. имеющее одинаковые имя, тип и размер, необходимо для установления взаимосвязи между родительской и дочерней таблицами. Таблицы могут быть объединены параллельно, последовательно и смешанно.

Перед установлением взаимосвязей все таблицы следует открыть в своих рабочих областях. Таблицы попарно должны иметь хотя бы одно общее поле, для которого в обеих таблицах (или хотя бы в одной) должен быть построен индекс.

Для организации взаимосвязей «один к одному», «один ко многим» используют различные команды.

В реляционной БД взаимосвязи между таблицами можно установить также с помощью главного меню. Для этого предварительно в каждой таблице строят первичный и внешние ключи. Затем выводят на экран диалоговое окно **Table Designer**. Указатель мыши наводят на имя первичного ключа родительской таблицы и перетаскивают его внутрь дочерней таблицы до имени соответствующего внешнего ключа. Обратная буксировка (от дочерней таблицы к родительской) не допустима. Проверить, а при необходимости и уточнить параметры взаимосвязи можно с помощью диалогового окна **Edit Relationship (Редактировать связи)**. На рис. 9.5 представлен пример схемы взаимосвязи таблиц, образующих БД, по ключевым полям (индексам).

Имеются команды для сведения информации из нескольких таблиц в одну, для корректировки данных в связанных таблицах, для создания итогового табличного файла (содержит суммы по указанным полям) и др.

Меню является основным инструментом диалога в БД. В FoxPro можно создать меню различных типов: световое меню типа FOX и типа dBase, кнопочное меню.

Световое меню типа FOX существует в трех вариантах:

- произвольного типа — LIGHTBAR-меню;
- вертикальное — POPUP-меню;
- двухуровневое — PULLDOWN-меню.

Световое меню типа dBase существует в двух вариантах:

- вертикальное — POPUP-меню;
- горизонтальное — BAR-меню;

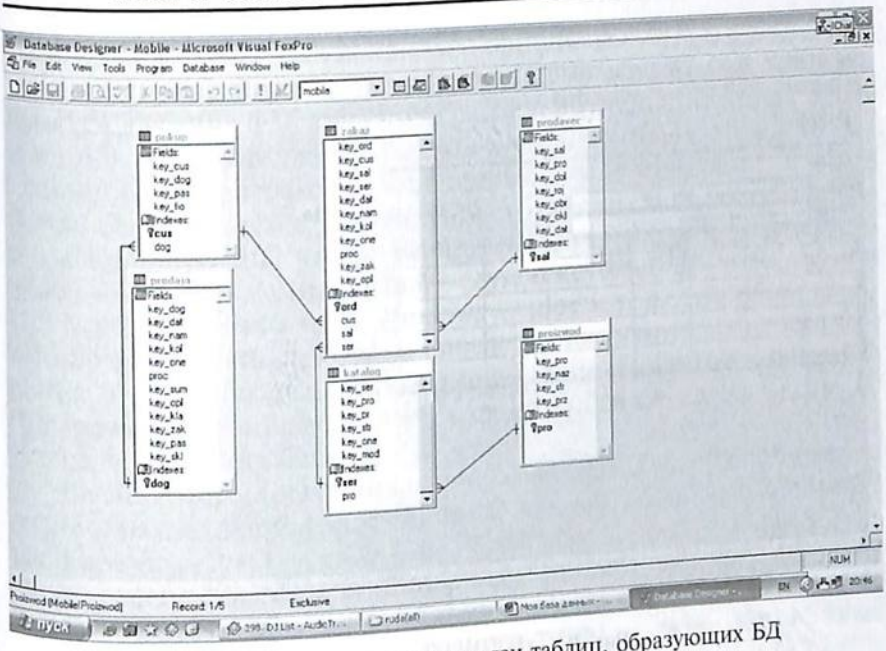


Рис. 9.5. Пример схемы взаимосвязи таблиц, образующих БД

Кнопочное меню представляет собой набор одно- или двухкнопочных команд. Для создания кнопочного меню используются определенные команды.

Для физически существующей таблицы можно создать экранную форму с помощью Мастера форм (Form Wizard) или Конструктора форм (Form Designer). Примерный вид готовой формы представлен на рис. 9.6.

Создать отчеты можно с помощью Мастера отчетов (Report Wizard) или Конструктора отчетов (New Report).

Мастер отчетов позволяет создать отчет по данным одной или нескольких таблиц. Допускается произвольный выбор полей, сортировка и группировка данных, изменение стиля отображения данных. Этапы создания отчета: определение окружения; размещение текста; размещение полей, линий, рисунков; перемещение объектов; сохранение отчета.

О **файлах операционной системы** см. раздел 2.8.

Файловая система определенным образом организована и включает в себя таблицу содержания, таблицу размещения файлов, таблицу определения файлов и т. п., а также область данных. ОС поддерживают обычные файлы, каталоги, специальные байт-ориентированные и блок-ориентированные файлы.

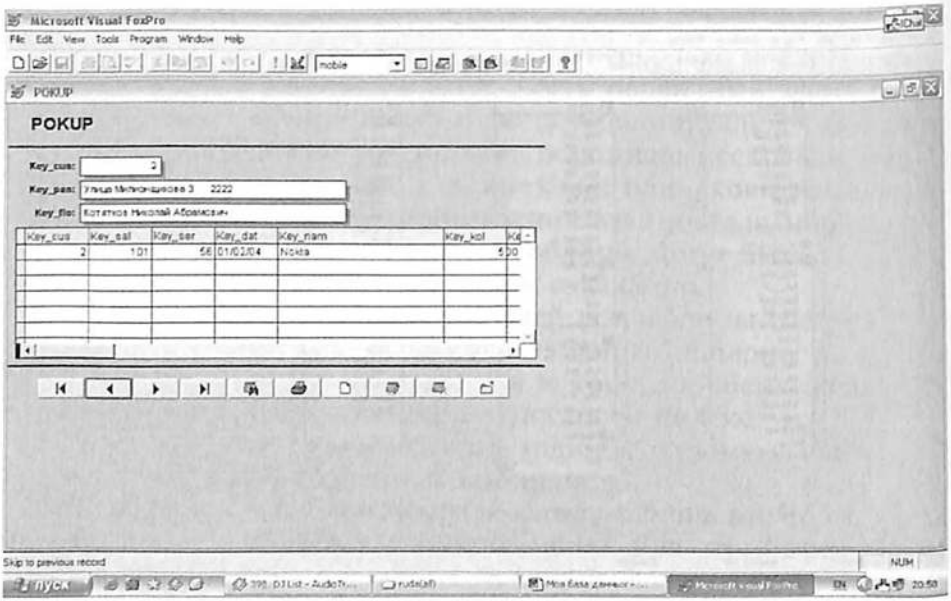


Рис. 9.6. Пример готовой формы

По способу выборки информации выделяют файлы последовательного, прямого доступа и доступа по индексу.

Форматная база включает типы, форматы и структуры данных, записей, документов.

Тип данных определяется типом значений, которые могут принимать данные (константы, переменные, выражения), и множеством допустимых операций над ними. В языках программирования, СУБД, ИПС и др. используют следующие основные типы данных: целый (`integer`), действительный (`real`), двоичный (`binary`), логический (`boolean`), символьный (`char`), длинный текстовый (`memo`), дата (`date`), время (`time`) и др.

Структуры:

- данных — элементарные данные, массивы;
- записей — агрегаты данных (именованные группы переменных разного типа);
- документов — файл (совокупность данных об объектах, при этом данные могут быть структурированы в элементарные или агрегатные (групповые) поля).

Основные структуры: массивы, записи, текстовые поля, множественные поля записи, групповые поля записи, повторяющиеся поля записи.

Формат — вид записи данных, организованных в файлы. Тип и назначение файла определяют его формат. В ОС, как правило, имеется два типа файлов: символьные и двоичные. В символьном виде обычно представлены исходные данные и программы, а в двоичном — программы, записанные в машинных кодах. Тип файла обозначается в расширении его имени. Например, исполняемые программы имеют расширение .exe и .com, командный файл — .bat, системный файл — .sys, текстовый файл в формате ОС MS-DOS — .txt, а ОС Windows — .doc и т. д. Текстовые файлы программ, написанных на определенном языке программирования, имеют и определенное расширение. Например, для программ на языке ассемблера — .asm, на Бейсике — .bas, на языке Паскаль — .pas, на Си — .c.

В настоящее время в различных системах имеется большое разнообразие форматов файлов. Они могут иметь следующие расширения:

- графические файлы (бинарные) — .bmp, .pcx, .tif, .gif, .jpg;
- табличные файлы (форматированы по столбцам и строкам, содержат информацию разного типа) — .wks, .xls, а также файлы реляционных баз данных — .dbf;
- звуковые файлы — .wav, .mid, .mp3, .mod;
- видеофайлы — .avi, .mov, .mpg;
- мультимедиа-файлы (содержат информацию разного вида: аудио, видео и др.) — .mpeg и др.

Для обмена информацией между системами существуют коммуникативные (обменные) форматы:

- МЕКОФ, карточный и др. — для обмена библиографической информацией;
- MARC — для обмена библиотечно-справочной информацией;
- HTML, SGML, RTF, ODA и др. — для обмена полнотекстовой документальной информацией.

При использовании этих форматов следует учитывать необходимость правильного распознавания и «привязывания» данных, переносимых из одной системы в другую, а также обнаружения ошибок считывания и пропуска данных.

С этой целью применяют специальные методы записи данных: постоянной и переменной длины. При использовании метода записи постоянной длины строго соблюдают на носителе постоянство предусмотренной длины записи, используют разделители, метки, идентификаторы, могут в первом байте каждой

записи указывать ее длину и т. д. Для записей переменной длины ограничителем является распознаваемая устройством физическая метка.

Унифицированные документы, используемые в АСУ, представляют собой набор форм организационно-распорядительной информации в соответствии со стандартом и форм для внутрисистемного пользования.

Лексическая база включает классификаторы, кодификаторы, словари, тезаурусы и другие лексико-лингвистические таблицы. Они используются для ввода, обработки и вывода информации в АИС.

Классификатор — систематизированный свод (документ, словарь), отображающий закон разбиения множества объектов на классы, группы и т. д. Классификатор — это также свод кодовых обозначений классификации, наименований этих объектов и фасетов классификационных признаков объектов.

Кодификатор — словарь, в котором присвоены коды (символы) обозначаемым понятиям, объектам, сущностям и т. д.

Тезаурус — толковый словарь дескрипторов (слов или словосочетаний), значение которых объясняется через связи с другими лексическими единицами (дескрипторами).

Во-первых, следует понять, для чего создаются классификаторы, а во-вторых, рассмотреть их структуру и содержание. Очевидно, что огромный объем информации создается в сфере хозяйствования, т. е. производства. Поэтому очень важно представлять себе, что такое экономическая информация и как рационально ее организовать для использования в различных целях.

Экономическая информация — частный случай информации, относящейся к экономике. Это совокупность сведений, отражающих условия, состояния и результаты общественного производства.

Для управления экономикой используются планирование, учет, контроль, регулирование, диспетчеризация и т. д. При этом техническая, технологическая, социальная и любая другая информация становится экономической, так как применяется в экономике. Совокупность таких данных, используемых с указанной целью, образует *систему экономической информации.*

Экономическая информация обычно носит дискретный характер и представляется буквами, цифрами, символами. Может выражаться графиками, диаграммами и другими способами. На-

пример, применительно к среднему предприятию можно рассматривать постоянную и непостоянную (не всегда участвующую в процессах или производную) информацию.

Состав постоянной экономической информации включает следующую информацию: нормативную, расценочную, справочную, структурную, табличную, маршрутную.

К *нормативной информации* относятся: конструкторско-технологические нормативы на оборудование и производственные помещения, сырье, материалы, трудовые и стоимостные нормативы, нормативы складских запасов материалов и многие другие. *Нормативы* регламентируют состав и количественные характеристики продукции, материалов и т. д., определяют порядок выполнения работ, указывают состав технических средств и состав исполнителей. Нормативно-справочная информация заимствуется в справочниках и нормативных документах.

Расценочная информация — данные о стоимости единицы продукции, материальных ресурсов, выполняемых работ. Расценочная информация утверждается соответствующими организациями.

Справочная информация — информация об относительно постоянных свойствах объектов (табельный номер сотрудника, номер цеха и др.).

Структурная информация характеризует структурный состав объекта, т. е. последовательность вхождения каждого составляющего элемента в некоторое соединение (например, в автомобиле 40 000 деталей, а данный завод изготавливает 20—25 деталей).

Табличная информация — совокупность данных, полученных в результате вычисления по соответствующим формулам и занесения в таблицу значений переменных и выходных данных — результатов вычислений (например, таблица налогов по заработной плате и т. п.).

Маршрутная информация — отражает технологическую последовательность выполнения процесса (обработки деталей, очередности сборки и т. д.).

Вся эта информация должна быть рационально организована, структурирована, т. е. классифицирована.

Классификация — система распределения всего множества предметов или понятий по общим признакам или различиям на отдельные группы или подмножества.

В СССР действовала система классификации экономической информации, включающая десятки миллионов наименований

изделий и конструкторской документации, 1,5 млн предприятий и организаций, 10 млн материальных и трудовых нормативов, 1 млн стандартов и технических условий, 70 000 показателей финансов, бухгалтерской, статистической и производственной технико-экономической информации.

К настоящему времени в нашей стране разработаны общероссийские классификаторы (ОК). Издано Постановление Правительства РФ от 10 ноября 2003 г. «Об общероссийских классификаторах технико-экономической и социальной информации в социально-экономической области». Ниже приводится перечень некоторых из них:

1. ОК информации об общероссийских классификаторах (ОКОК).
2. ОК стандартов (ОКС).
3. ОК объектов административно-территориального деления (ОКАТО).
4. ОК предприятий и организаций (ОКОП).
5. ОК органов государственной власти и управления (ОКОГУ).
6. ОК экономических регионов (ОКЭР).
7. ОК видов экономической деятельности (ОКВЭД).
8. ОК занятий (ОКЗ).
9. ОК управленческой документации (ОКУД).
10. ОК продукции (ОКП).
11. ОК информации о населении (ОКИН).
12. ОК информации по социальной защите населения (ОКИСЗН).
13. ОК услуг населению (ОКУН).
14. ОК профессий рабочих, должностей служащих и тарифных разрядов (ОКПДТР).
15. ОК валют (ОКВ).
16. ОК основных фондов (ОКОФ).
17. ОК изделий и конструкторских документов (ЕСКД).
18. ОК единиц измерения (ОКЕИ).
19. ОК специальностей по образованию (ОКСО).
20. ОК начального профессионального образования (ОКНПО).
21. ОК специальностей высшей научной квалификации (ОКСВНК).
22. ОК полезных ископаемых и подземных вод (ОКПИиПВ).
23. ОК гидроэнергетических ресурсов (ОКГР).
24. ОК стран мира (ОКСМ).
25. ОК форм собственности и др.

При классификации множества выбирается определенная *система кодирования*, т. е. каждой позиции классификатора присваивается кодовое обозначение.

Кодирование — это перевод записи из одной формы в другую с помощью символов, процесс, при котором информация, выраженная в одной системе знаков, представляется в другой системе. Кодирование осуществляется для экономии памяти, времени для заполнения форм, ввода и обработки документов, минимизации ошибок. Например, фамилию можно кодировать символом 1 и т. д.

Система кодирования — совокупность правил, определяющих систему знаков и порядок их использования в работе с информацией.

Кодирование должно обеспечить:

- компактное и удобное представление информации;
- удобство обработки информации;
- сопряжение различных технических устройств и людей, работающих с кодированной информацией.

Системы кодирования разных уровней должны быть увязаны. Эта проблема может решаться с помощью машинного перекодирования, машинной подготовки данных для управления кодированием.

Системное кодирование может быть:

1. Последовательное — реализуется иерархическая система классификаций.

2. Параллельное — соответствует фасетной классификации. Отсутствует взаимная зависимость между группами, ее легко изменять.

3. Серийно-порядковое — при наличии двух классификационных признаков объекты делят на группы. В каждой группе — порядковая система кодов, в каждой группе — серия.

Принципы кодирования:

- выбирать значимость кода минимальной;
- обеспечивать максимальную логичность кода;
- учитывать особенности применяемых технических средств;
- учитывать особенности автоматизируемого производства;
- использовать существующие коды номенклатур и общих обозначений;
- учитывать перспективы развития производства.

Коды могут быть разного вида и сочетания (рис. 9.7).

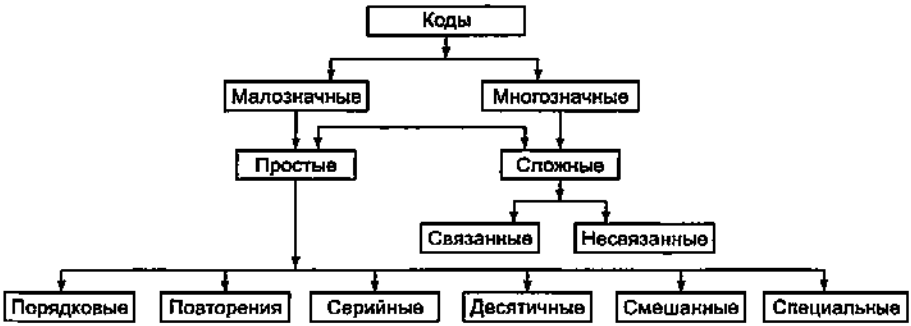


Рис. 9.7. Классификация цифровых кодов технико-экономической информации

В нашей стране действует ЕСКК — Единая система классификации и кодирования.

Для классификации экономической информации используют десятичную систему обозначений для выбираемых группировок признаков (рис. 9.8).

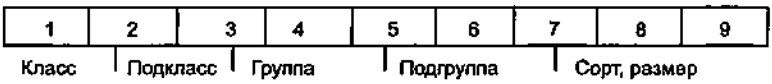


Рис. 9.8. Система обозначений для классификации экономической информации

Несмотря на свои преимущества, общероссийские классификаторы громоздки, шифры избыточны. Поэтому в ряде случаев целесообразно разрабатывать собственные (локальные) классификаторы и соответствующие им шифры.

Шифр — условное обозначение определенного информационного понятия. Шифр может состоять из комбинации букв, чисел или букв и чисел. Шифры должны:

- обеспечить возможность точной идентификации объекта;
- иметь по возможности меньшее число разрядов;
- шифр одного и того же объекта должен быть одинаков во всех решаемых задачах;
- построение шифра должно быть оптимальным.

Например, коды органа старшего уровня образуют из действующих кодов. На уровне министерства коды образуют из них же с приставкой кодов министерства. На уровне предприятия коды образуют из них же с приставкой кодов предприятия. Это упрощает процесс перекодирования.

Например, Общероссийский классификатор специальностей по образованию ОКСО (дата введения — 7 января 1994 г.) являлся составной частью ЕСКК РФ. Он предназначался для использования в процессе автоматизированной обработки и обмена информацией на всех уровнях управления хозяйством страны. Объекты классификации ОКСО — группы специальностей, области знаний, специальности среднего профессионального и высшего образования, направления подготовки в высшем образовании, специализации. Классификатор состоит из двух разделов: «Среднее профессиональное образование» и «Высшее образование». Описание объекта классификации включает три блока: идентификации, наименования и фасетов классификационных признаков.

Блок идентификации строится по иерархическому методу классификации и использует серийно-порядковый, последовательный и параллельный методы кодирования. Длина кода — 7 десятичных знаков и контрольное число (КЧ). Формула для составления кода имеет вид ХХ+ХХ+ХХ+Х+КЧ, где, например, для специальностей среднего профессионального и высшего образования первый и второй знаки — группа специальностей, третий и четвертый знаки — специальность, пятый и шестой знаки — специализация, седьмой знак — уровень подготовки. Для седьмого знака принято следующее: 1 — специальности среднего профессионального образования, 2 — направления высшего образования, 3 — специальности высшего образования. Блок наименования содержит наименование группы специальностей, области знаний (наук), специальности, направления подготовки или специализации.

Блок фасетов классификационных признаков указывает на группу образовательных программ по МСКО (Международной стандартной классификации образования ЮНЕСКО), включает 5 признаков и строится по формуле Х+ХХ+ХХ. В формуле первый знак — ступень образования, второй и третий знаки — изучаемая область в рамках ступени, четвертый и пятый знаки — группа образовательных программ в пределах изучаемой области.

Ниже приведен фрагмент ОКСО, классифицирующий специальности среднего профессионального образования (табл. 9.1).

Тезаурус включает, как правило, лексико-семантические алфавитные указатели и систематические указатели. Лексико-семантические алфавитные указатели содержат дескрипторы с сопровождающими дескрипторными статьями и недескрипторы (с отсылкой «см.» к соответствующему дескриптору).

Таблица 9.1. **Фрагмент Общероссийского классификатора специальностей по образованию ОК 009—93**

Код	КЧ	Наименование	Код по МСКО
0100 00 1	9	**Естественно-научные специальности**	
0101 00 1	2	**Гидрология**	54299
	6	Метеорология	54299
0200 00 1	0	**Гуманитарно-социальные специальности**	
0201 00 1	4	Правоведение	53800
0202 00 1	8	Право и организация социального обеспечения	58912
0203 00 1	1	Правоохранительная деятельность	58912
0204 00 1	5	Криминалистическая экспертиза	58912
0205 00 1	9	Социальная работа	58932
0206 00 1	2	Издательское дело	58409

Дескрипторная статья — упорядоченный класс дескрипторов и недескрипторов (синонимов дескрипторов), связанных внеконтекстными отношениями с заглавным дескриптором.

Пример дескрипторной статьи из тезауруса КАДИПС (картографической автоматизированной документальной ИПС) представлен ниже.

499 ДОБЫВАЮЩАЯ ПРОМЫШЛЕННОСТЬ
- с горная промышленность
  горнодобывающая промышленность
- в ПРОМЫШЛЕННОСТЬ (496)
- н ДОБЫЧА МЕТАЛЛИЧЕСКОГО СЫРЬЯ (500)
  ДОБЫЧА НЕМЕТАЛЛИЧЕСКОГО СЫРЬЯ (503)
  ТОПЛИВОДОБЫВАЮЩАЯ ПРОМЫШЛЕННОСТЬ (507)
- а ПОЛЕЗНЫЕ ИСКОПАЕМЫЕ (72)

В дескрипторной статье введены следующие обозначения:
499, (503) и др. — коды (цифры в скобках или без скобок);
с — синоним;
в — выше (вышестоящее, подчиняющее понятие);
н — ниже (нижестоящее, подчиненное понятие);
а — ассоциация (обозначение ассоциативной связи).

Систематические указатели содержат дескрипторы, сгруппированные по тем или иным классификационным признакам (например, по содержанию, календарным периодам и т. д.).

Языковые средства — информационные языки (языки описания данных — ЯОД, словари данных, языки манипулирования данными — ЯМД).

ЯОД — язык высокого уровня, предназначен для формализованного описания типов данных, их структур и взаимосвязей. С его помощью администратор БД и программисты описывают структуру и содержимое БД.

Словарь данных — файл или таблица БД, содержащие описания данных и типы их обработки.

ЯМД предназначен для поиска и отображения данных и включает в себя язык запросов (ЯЗ) и информационно-поисковый язык (ИПЯ). ЯМД является средством, которое применяется пользователем или прикладным программистом для выполнения операций над данными.

Компоненты запросов хорошо развиты в языках SQL, FoxPro и др. В настоящее время стандартом стал язык SQL (Structured Query Language), в котором предусмотрены компоненты для описания БД.

ИПЯ содержат лексику (словарь) и грамматику. Единицы текста словаря используют для индексирования, а по правилам грамматики составляют ПОД — поисковые образы документов (фактов).

Различают индексирование предкоординируемое и посткоординируемое. Первое предполагает наличие классов, к которым следует относить закодированный соответствующим индексом документ. Второе позволяет приписать документу несколько дескрипторов. Например, к ИПЯ первого вида можно отнести классификационную систему УДК. В основе ИПЯ второго вида лежат тезаурусы, они являются ИПЯ дескрипторного типа.

Контрольные вопросы

1. Каково определение и принципы создания информационного обеспечения?
2. Каковы этапы создания ИО?
3. Каков состав информационного обеспечения АИС?
4. Что такое структуризация информации и как она структурируется в базах данных?
5. Что такое форматная база?

6. Что такое лексическая база?
7. Что такое Единая система классификации и кодирования информации? Приведите примеры общероссийских классификаторов.
8. Для чего предназначены ЯОД и ЯМД?

9.2. Математическое обеспечение

МО в АС предназначено для реализации управляющих решений, рассматриваемых как совокупность действий для достижения поставленных целей в рамках технического задания.

Состав МО:

1. Математическое описание (формализация) задач.
2. Математические модели и их оптимизация.
3. Данные, подготовленные для описания исследуемых процессов.
4. Алгоритмы решения задач.
5. Анализ моделей и алгоритмов по результатам выполненных работ на ЭВМ.

МО АС должна выполнять следующие функции:

- реализацию любых процедур обработки данных;
- компоновку рабочих программ решения конкретных задач из стандартных программ и оригинальных блоков;
- организацию управления процессом решения задач и их комплексов;
- реализацию экономико-математических методов решения оптимизационных задач. МО АС должна содержать средства автоматизации программирования задач, а также средства компоновки рабочих моделей конкретных систем из стандартных программ и их обслуживания.

В МО по последовательности проектирования АСУ рассматривает три уровня:

1) математическое обеспечение конкретной АС, которым определяется ее мощность;

2 автоматизация проектирования АС;

3) автоматизация программирования и организация работ на ЭВМ.

Разработка МО предполагает выполнение следующих этапов:

- создание модели системы;
- разработка укрупненного алгоритма;
- разработка алгоритмов отдельных элементов МО;

- проверка достоверности алгоритмов (выбор вычислительных средств, проведение программирования, проверка достоверности программы).

Прежде всего выполняют постановку задачи моделирования. При этом проводят:

- определение требований к исходной информации, ее сбор;
- выдвижение гипотез и предположений;
- определение параметров и переменных модели;
- обоснование выбора показателей и критериев эффективности системы;
- определение содержания и описание модели (основной документ).

АСУ является *информационной системой*, поэтому ее функционирование есть последовательность действий по обработке информации, предназначенной для управления. Рассмотрим структуру МО на примере АСУ. МО АСУ включает совокупность методов и средств, позволяющих строить экономико-математические модели задач управления объектом: методы + модели + алгоритмы обработки информации.

Общее МО реализуется в виде программ операционной системы (ОС) и системы программирования (СП) — для оптимизации процесса программирования и диагностики программ. В состав МО системы входят средства, документация и методы.

К первым относятся средства моделирования, описания задач управления, методы оптимизации моделей и математической статистики. Ко второй относится документация описания задач, заданий на алгоритмическом языке экономико-математической модели, алгоритмов решения задач, контрольных примеров, методов определения типа задач, методов оценки вычислительной сложности алгоритмов, методов оценки решений.

Основным фактором успешного решения задач является научно обоснованная *формализация задачи*. Наиболее трудным является формализация задач на уровне *спецификаций*, когда необходимо содержательное представление задачи перевести в формальное описание. Решение формализованной задачи позволяет получить четкие оценки ожидаемых результатов. Формализация успешно осуществляется на основе математического моделирования, которое является неотъемлемой частью науки управления, успешно реализуемой в рамках ИСО[1].

---

[1] ИСО — метод (способ) исследования операций.

Существует множество различных типов моделей: физические, аналоговые, интуитивные и т. д. Особое место среди них занимают математические модели (ММ), которые, по мнению академика А. А. Самарского, и являются самым большим достижением научно-технической революции XX в.

Модель — это информационный образ реального объекта, воспроизводящий данный объект (систему) с определенной степенью точности и в форме, часто отличной от формы самого объекта.

Назначение модели — поиск значений управляемых переменных, оптимизирующих критерий эффективности операций.

Создание моделей реальных бизнес-проектов и объектов управления является высшей точкой операционного подхода к решению задач информационного управления.

Модель позволяет выявить альтернативы решения задачи и оценить результаты, к которым они приводят, определить данные, необходимые для оценки имеющихся альтернатив. Это обеспечивает получение обоснованных выводов. Модель является средством формирования четкого представления о действительности:

1. Модель может быть физической копией реального объекта. В таких случаях говорят о физическом моделировании, физических моделях (копии самолетов, автомобилей — уменьшенные или увеличенные). Их свойства близки свойствам реального объекта, а стоимость гораздо меньше.

2. Аналоговые модели — аналог исследования объекта, в той или иной форме воспроизводящий функции реального объекта (график, описанная связь между величинами).

3. Математические модели (ММ) — совокупность математических объектов (чисел, символов, множеств и т. д.) и связей между ними, отражающих в символьной форме важнейшие для исследования свойства объекта.

Так, формула $P = \frac{n_p}{U}$ (где P — уровень рентабельности, n_p — прибыль, U — издержки производства) — математическая модель, описывающая одно из важнейших свойств действующего предприятия.

Семантические модели отражают функции исследования объекта в виде семантических алгоритмов (правил, свойств, признаков), описанных в словесной форме.

Моделирование — способ системного анализа проектирования, при котором используют математические или физические модели функционирования всей системы или ее части. Полнота и реальность модели зависят от тех вопросов, на которые надо ответить, степени изученности системы, а также среды ее функционирования.

Математическое моделирование — важнейший трудоемкий и наукоемкий процесс при создании и сопровождении сложных автоматизированных информационных систем, который позволяет в должной степени оценить вероятности успеха, связанные с этим риски, прибыли и ущербы. В результате правильного моделирования углубляются и моделируются знания о системе, о связи возможных результатов с различными характеристиками этой системы, условиях создания и функционирования, и в конечном итоге устанавливается приемлемая степень достижения целей, которые перед ней ставились. Все вышеперечисленное позволяет заказчику правильно и доказательно сформулировать требования технического задания (ТЗ), разработчику — рационально их выполнить без излишних затрат ресурсов, а пользователю — максимально эффективно реализовать на практике заложенный потенциал системы.

Математическое моделирование — процесс создания математических моделей и оперирование ими с целью получения требуемых сведений о реальном объекте. Математическая модель должна отражать сущность моделируемой проблемы управления.

Последовательно осуществляют разработку математической модели (ММ) и ее машинную реализацию:

1) построение концептуальной модели;
2) разработку алгоритма модели системы;
3) разработку программы реализации модели системы;
4) проведение машинных экспериментов с моделью системы.

К математическим моделям предъявляются требования универсальности, адекватности и экономичности (меньше затрат ресурсов).

Структурные математические модели отображают структурные свойства объекта, топологические и геометрические.

В топологических моделях отображаются состав и взаимосвязи элементов объекта. Топологические модели могут иметь форму графов, таблиц (матриц), списков и т. п. (в транспортной системе — расписания и т. п.).

Функциональные математические модели предназначены для отображения процессов (физических или информационных), протекающих в объекте при его функционировании или изготовлении. Обычно функции модели содержат алгоритмы, связывающие переменные, внутренние, внешние или выходные параметры. Выделение аспектов описания позволяет выделять комплексы алгоритмов, относящихся к той или иной деятельности объекта, и приводят декомпозицию системы к определенному признаку.

Деление объектов на иерархические уровни приводит к определенным уровням моделирования. В зависимости от места и иерархии описаний математические модели делятся на микро-, макро- и метамодели. Эти модели по своей структуре и содержащихся в них математических объектах могут не различаться, что позволяет применять одинаковые алгоритмы их решения. Различие состоит в том, что на более высоком уровне компоненты модели принимают вид сложных совокупностей элементов предыдущего уровня. Этими же аспектами определяется и разделение моделей по степени детализации описаний объектов.

По способу представления свойств объектов выделяют следующие математические модели: аналитические, алгоритмические, имитационные, семантические.

Автоматизированное проектирование оптимальных объектов и систем на основе математических методов с использованием компьютеров содержит две основные задачи:

- разработку математической модели объекта, содержащей все основные технико-экономические требования к создаваемому объекту (работоспособность, технологичность, допустимая стоимость и т. п.);
- организацию такого вычислительного процесса, который автоматизирует выполнение всех требований математической модели.

В исследованиях операций, как правило, используются математические модели.

Операционная модель — это совокупность алгоритмов, описывающих функциональные свойства проектируемого объекта, отвечающего всем требованиям, предъявляемым в рамках конкретных задач проектирования. Операционная модель выражает зависимость критерия эффективности операции от параметров, выбранных оперирующей стороной, а также условий проведения операций. Функционально это выражается зависимостью $W = F^*(A_j; X_j)$, где W — выражение критерия эффективности опера-

ции; F — оператор (символ модели); A_j — информация, вводимая в модель, на которую оператор не оказывает влияние; X_l — управляемые параметры.

В *аналитических моделях* критерий связан с величинами A_j и X_l математическими зависимостями, по которым можно определить экстремальное значение либо непосредственно, либо с помощью численных методов на ЭВМ. Связь между W и X_l и A_j может быть очень сложной.

Общих математических методов нахождения экстремума функции любого вида при произвольных ограничениях не существует. Но для целевой функции и системы ограничений, обладающих определенными свойствами, имеются специальные методы, исследуемые математическим программированием.

Под общей задачей математического программирования понимают задачу отыскания глобального экстремума функции f_n переменных $x_1, \ldots, x_n$ на множестве M n-мерного пространства. При этом функция $f(x)$ называется целевой функцией, а множество M обычно задается с помощью уравнений и неравенств следующего вида:

$$g_i(x) = 0,\ i = 1, \ldots, k;$$

$$g_i(x) \le 0,\ i = k + 1, \ldots, l;$$

$$g_i(x) \ge 0,\ i = l + 1, \ldots, m,$$

а также с помощью разного рода дополнительных условий, здесь $g_i(x) = g_i(x_1, \ldots, x_n)$ — известные функции n переменных.

Существующие ММ экономических систем можно представить тремя группами:

1. Уравнения второй или третьей степени (алгебраические).
2. Модели систем массового обслуживания (статистические).
3. Модели больших и очень больших систем.

Алгебраическое моделирование — процесс функционирования системы во времени, причем имитируются элементарные явления с сохранением их логической структуры и последовательности протекания во времени.

Статистические модели строятся методом случайных статистических чисел.

Одним из важнейших видов математического моделирования является вероятностное (статистическое) моделирование, когда используется вероятностное подобие. Вероятностные модели определяют средний суммарный результат, получающийся от дей-

ствия многих случайных факторов. В модели с помощью случайных чисел имитируется действие неопределенных и случайных факторов.

Моделирование больших и очень больших систем прежде всего выполняют с помощью *алгоритмического моделирования*, которое описывает процесс функционирования системы во времени. При этом имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания во времени.

Для разработки *укрупненного алгоритма* выполняют построение логической схемы алгоритма модели системы с учетом математических соотношений, полученных при формализации задач. При этом выполняются:

- разработка структуры массивов информации;
- определение для каждого массива носителя информации;
- разделение процедуры решения задачи (комплекса задач) на отдельные самостоятельные элементы;
- разработка укрупненного алгоритма из выделенных самостоятельных элементов.

Разработка *алгоритмов отдельных элементов МО* предполагает построение подробных структурных схем машинной реализации всех составляющих МО. Предусматриваются различные реакции на разные сбойные ситуации.

После алгоритмизации осуществляют программирование с помощью выбранных вычислительных средств, проверку достоверности программы. Достоверность алгоритма проверяют путем реализации его с помощью программ на ЭВМ. Отладка выполняется на конкретном примере. Далее проводят опытную эксплуатацию в различных ситуациях. При необходимости осуществляют доработку МО.

Документация на МО должна полностью содержать то, что должен знать заказчик, в ясном и квалифицированном изложении. Поэтому при сдаче системы в эксплуатацию заказчику передают необходимую документацию по МО на машинных носителях с программами обучения персонала работе с документами и системой в соответствии с инструкцией по эксплуатации. При эксплуатации системы ведется авторский надзор, выполняется совместный анализ полученных результатов и устранение ошибок.

Метод исследования операций (*ИСО*) занимается количественным обоснованием принимаемых решений, связанных с опти-

мальным управлением организационными системами в различных областях человеческой деятельности. ИСО играет большую роль при решении конкретных практических задач управления, разработке, создании и эксплуатации АСУ.

Рассматривая процесс управления как совокупность операций, его можно реализовать методами исследования операций, т. е. численно оценить различные варианты достижения цели, дать в руки управленцев основания для принятия решений, выраженные в числовой мере и представленные в сжатом виде. Методы ИСО имеют большое значение в автоматизации процесса выработки управляющих решений.

Особенности ИСО:

1. Количественное обоснование (квантификация) рассматриваемых вариантов решений. Обязательно учитывают три момента:

- полноту достижения цели (какой будет эффект?);
- цену (каковы будут затраты различных ресурсов?);
- степень риска (каковы могут быть потери?).

2. Системная методология. Все процессы рассматриваются в качестве систем, т. е. как целенаправленные, взаимодействующие совокупности элементов. Например, для расширения сферы применения ЭВМ требуется:

- создать современное МО управляющих ЭВМ (широкий набор стандартных программ, трансляторы с различных языков, развитые ОС, системы отладки программного обеспечения);
- подготовить и формализовать, т. е. математически описать, задачи в тех областях человеческой деятельности, где предполагается использовать управляющие ЭВМ;
- разработать и создать периферийные устройства связи ЭВМ с внешним миром (специализированные линии связи, устройства автоматического ввода/вывода и отображения информации);
- подготовить необходимые кадры и т. д.

3. Из-за невозможности осуществления физического эксперимента с изучаемой системой, так как она сложна и дорого стоит, применяют математическое моделирование систем (ММС). ММС представляет собой некоторое формализованное описание, находящееся в определенном сходстве с исследуемыми процессами или системами. Познавая ее свойства, можно познать и свойства отраженной моделью реальной системы.

С помощью модели решают следующие задачи, связанные с реальной системой:

- описание поведения системы;
- объяснения;
- предсказание.

4. Огромный объем вычислений при получении решений с помощью моделей. Учет десятков тысяч факторов.

5. Рекомендательный характер. Цель ИСО — оказание помощи лицу, ответственному за принятие решения.

6. ИСО синтезирует достижения математики (особенно следующих ее разделов: математическое программирование, теория игр, теория вероятностей, математическая статистика и др.) и неформальных методов в практике подготовки управляющих решений (метод экспертных оценок, имитационное моделирование, операционные игры и т. п.).

ИСО рассматривает операцию как упорядоченную совокупность взаимосвязанных действий, объединенных единым замыслом и направленных на достижение определенной цели.

Задача — это желаемый результат деятельности, достижимый за намеченный промежуток времени и характеризующийся некоторым набором количественных показателей.

Цель — более общая категория, она становится задачей, если указан срок и количественные показатели результата. Цель достигается в результате решения задач.

Оперативное управление состоит в контроле за ходом операции, устранении отклонений от запланированного течения операций. Под эффективностью операции понимают степень или полноту достижения цели. Количественной мерой эффективности является критерий эффективности. Он выбирается в зависимости от характера задачи. В задачах экономического характера это прибыль, рентабельность, себестоимость выпускаемой продукции, затраты на содержание предприятий и т. д.; в технологических задачах — производительность оборудования, его долговечность, КПД, характеристики надежности и т. д.

Для получения обобщенного критерия эффективности существует много способов. Наиболее распространенный — объединение частных критериев с помощью весовых коэффициентов:

$$W = a_1 W_1 + a_2 W_2 + \ldots + a_i W_i,$$

где a_i — положительные или отрицательные коэффициенты веса частных критериев W_i (как правило, положительные коэффици-

енты требуют максимизации, а отрицательные — минимизации). Часто добавляется условие нормировки, требующее, чтобы сумма весовых коэффициентов была равна единице:

$$\sum_{i=1}^{n} |a_i| = 1.$$

Определение весовых коэффициентов — трудоемкая задача. Решение ее достигается экспертным опросом, логическим анализом либо статистическим моделированием.

Существует много методов определения критериев эффективности:

1. Лапласа (когда все условия равновероятны).
2. Вальда (критерий осторожного поведения или макс-минный критерий).
3. Гурвица (критерий компромиссного поведения).
4. Сэвиджа (критерий минимаксного риска) и др.

Критерии эффективности описаны во многих источниках.

Когда нет другой возможности, используют ранговый подход. Ранг — количественная оценка критерия эффективности, носящая субъективный характер, так как качественному признаку ставится в соответствие некоторое число.

В АСУ часто применяют методы ИСО, для которых наиболее типичны следующие классы оптимизации задач:

1) распределения и назначения;
2) упорядочения;
3) массового обслуживания;
4) управления запасами;
5) об износе и замене оборудования;
6) выбора маршрута и проектирования сетей;
7) состязаний;
8) поиска.

Метод линейного программирования используют для поиска оптимальных плановых решений. В АСУ его можно свести к двум основным постановкам задач:

1) получение запланированного эффекта при минимальных затратах;

2) получение максимального эффекта при использовании заданных организацией ресурсов.

Механизм экономических отношений описывается целым рядом взаимосвязанных показателей: товарооборот, рентабель-

ность, избытки обращения, ассортимент товаров, площадь торговых залов и подсобных помещений, количество квалифицированных работников, виды оборудования, товарные запасы, система обработки документов, форма обслуживания потребителей и т. д.

На величину этих показателей влияют такие факторы, как ритмичность, частота и объемы выпуска продукции, поставка ее заказчикам, ассортимент и качество продукции, наличие и исправность оборудования, количество персонала и т. д.

Все экономические показатели и факторы можно разделить на:

- неуправляемые (z_1, z_2, ..., z_i, ..., z_m);
- управляемые (x_1, x_2, ..., x_j, ..., x_n).

На этом основании целевую функцию можно записать в виде уравнения этих показателей с критерием оптимальности (extremum):

$$F = f(x_1, x_2, \ldots, x_j, \ldots, x_n; z_1, z_2, \ldots, z_i, \ldots, z_m) \rightarrow \text{extr}.$$

Постановка задачи завершается переводом задачи планирования с языка экономики на язык математики. Этот процесс связан с построением одного или нескольких математических уравнений или неравенств, которые в совокупности описывают функциональные связи критерия оптимальности с показателями ресурсов, факторами и неизвестными значениями управляемых показателей. Такая запись экономической задачи является экономико-математической.

Для представления экономической постановки задачи в виде математической модели линейного программирования необходимо целевую функцию представить в линейной форме, а связь с ограниченными ресурсами описать посредством линейных уравнений или неравенств. Кроме того, вводится дополнительное ограничение — значения переменных должны быть неотрицательны, т. е. $x_1 \geq 0$, $x_2 \geq 0$, ..., $x_j \geq 0$, ..., $x_n \geq 0$. В целом экономико-математическая формулировка и модель *общей задачи линейного программирования* (ОЗЛП) имеет следующий вид: найти максимальное (минимальное) значение линейной целевой функции:

$$F(\bar{x}) = \sum_{j=1}^{n} c_j x_j \leftarrow \max(\min) \qquad (1)$$

при условии следующих ограничений:

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j \le b_i, \; i = \overline{1, k}; & (2) \\ \sum_{j=1}^{n} a_{ij} x_j = b_j, \; i = \overline{k+1, m}, \; k \le m; & (3) \\ X_j \ge 0, \; j = \overline{1, e}, \; e \le n, & (4) \end{cases}$$

где a_{ij}, b_i, c_j — заданные постоянные величины.

Стандартной задачей линейного программирования называют задачу, которая состоит в определении максимального (минимального) значения целевой функции (1) при выполнении условий (2) и (4), где $k = 0$ и $e = n$.

Канонической (или основной) задачей линейного программирования называют задачу, которая состоит в определении максимального (минимального) значения целевой функции при выполнении условий (3) и (4).

Совокупность чисел $\bar{x} = (x_1, x_2, \ldots, x_n)$, удовлетворяющих ограничениям, называется допустимым решением (или планом).

План $\bar{x}^* = (x_1^*, x_2^*, \ldots, x_m^*)$, при котором целевая функция задачи принимает максимальное (минимальное) значение, называется *оптимальным*.

В случае, когда требуется найти минимальное значение функции

$$F(\bar{x}) = (c_1 x_1 + c_2 x_2 + \ldots + c_n x_n),$$

можно перейти к нахождению максимального значения функции

$$F_1(\bar{x}) = -F(\bar{x}) = -c_1 x_1 - c_2 x_2 - \ldots - c_n x_n,$$

так как $\min F(\bar{x}) = -\max F_1(\bar{x})$.

Ограничение «неравенство исходной ЗЛП, имеющее вид ≤», преобразуется в ограничение равенства с добавлением левой части дополнительной неотрицательной переменной. Ограничение «неравенство вида ≥» преобразуется в ограничение «равенство вычитанием из левой части дополнительной неотрицательной переменной».

Если ограничения задачи отражают наличие и равенство производственных ресурсов, то числовое значение дополнительной переменной в плане задачи, записанной в форме основной, равно объему неиспользуемого соответствующего ресурса.

Запишем в ОЗЛП ограничения (2, 3) в векторной форме:

$$X_1\bar{A}_1 + X_2\bar{A}_2 + \ldots + X_m\bar{A}_m = \bar{B}, \qquad (5)$$

где $\bar{A}_1$, $\bar{A}_2$, ..., $\bar{A}_m$ и $\bar{B}$ — m-мерные векторы-столбцы, составленные при неизвестных и свободных членах системы уравнений задачи.

План $\bar{x} = (x_1, x_2, \ldots, x_n)$ называется опорным планом ОЗЛП, если система векторов $\bar{A}_j$, входящих в разложение (5) с положительными коэффициентами x_j, линейно независима. Так как векторы $\bar{A}_j$ являются m-мерными, то из определения опорного плана следует, что число его положительных компонентов может превысить m.

Опорный план называется *невырожденным*, если он содержит ровно m положительных компонентов.

Если в опорном плане число положительных компонентов меньше m, то план является *вырожденным*.

Часто в практике рыночных условий приходится решать задачи оптимизации объемов выпуска продукции и расширения ее номенклатуры для сохранения достигнутого уровня прибыли в условиях насыщения рынка. Часто необходимо принимать управляющие решения по выбору направлений деятельности предприятий, обеспечивающих максимальную прибыль при ограниченных ресурсах. В этих целях применяют как простые способы (например, построение графиков), так и более сложные, используя расчеты.

Пример 9.1. Решение задачи графическим методом.

На предприятии выпускают два вида продукции: мотоциклы и велосипеды. Исходя из возможностей сборочного цеха, в нем могут собирать или 25 мотоциклов в день, или 100 велосипедов в день, либо комбинацию тех и других, определяемую приемлемыми трудозатратами. Склад может принять не более 70 изделий любого вида в сутки. Мотоцикл стоит в 2 раза дороже велосипеда. Требуется найти такой план выпуска продукции, который обеспечил бы предприятию наибольшую выручку.

Обозначим: X — число выпускаемых мотоциклов в день;

Y — число выпускаемых велосипедов в день;

T_1 — время (в часах) производства одного мотоцикла;

T_2 — время (в часах) производства одного велосипеда.

Из условия задачи следует, что $T_1 = 4T_2$.

Если завод работает круглосуточно, то при одновременном выпуске обоих изделий

$$T_1 \cdot X + T_2 \cdot Y \le 24,$$

или

$$4T_2 \cdot X + T_2 \cdot Y \le 24.$$

$4X + Y \le 24/T_2$, но $24/T_2$ — максимальное число производимых велосипедов = 100.

Итак: $4X + Y \le 100$.

Еще одно условие — ограниченная емкость склада:

$$X + Y \le 70.$$

Обозначим цену мотоцикла a_1 (руб.), цену велосипеда — a_2 (руб.).

По условию: $a_1 = 2a_2$.

Следовательно, общая цена дневной продукции S:

$$S = a_1 \cdot X + a_2 \cdot Y = 2a_2 \cdot X + a_2 \cdot Y = a_2(2X + Y).$$

Поскольку a_2 — заданная положительная константа, то наибольшего значения следует добиваться от величины $F = 2X + Y$.

Учитывая все условия задачи, приходим к ее математической модели, включающей систему неотрицательных целочисленных решений линейных неравенств

$$\begin{cases} 4X + Y \le 100; \\ X + Y \le 70. \end{cases} \qquad (*)$$

Проще всего задачу решить геометрически. Построим на плоскости (x, y) прямые, соответствующие неравенствам (*) и условию неотрицательности x и y. Координаты точки P (10; 60) — искомый *оптимальный* план производства при заданных ограничениях.

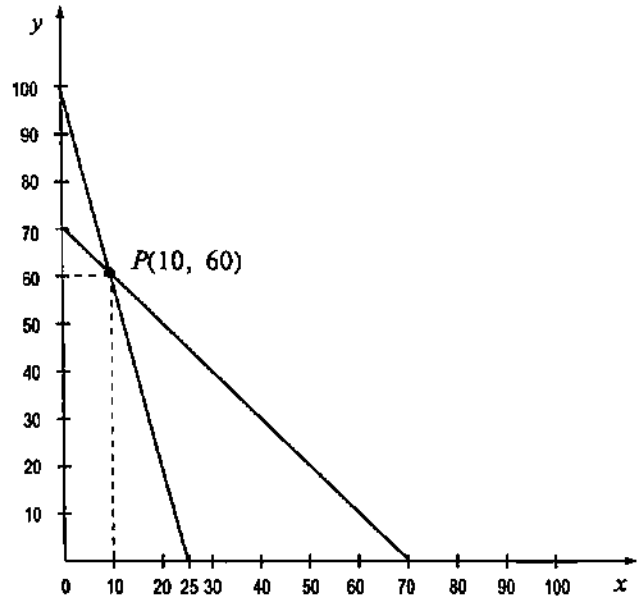


Рис. 9.9. Графическое решение задачи примера 9.1

Максимальное значение линейной функции

$$F = 2X + Y \qquad (**)$$

можно добавить при расширении возможностей склада.

Пример 9.2. Классическая транспортная задача (задача Хитчкока — Купманса) — о поставке грузов от поставщиков к потребителям — является типовой задачей для промышленных фирм, имеющих несколько предприятий, рынков сбыта и оптовых баз. Решение сводится к выбору оптимальных маршрутов, особенно когда фирмы ежемесячно пересматривают свои планы распределения продукции и номенклатура заказов меняется. Если нет других приоритетных целей, то задача состоит в том, чтобы минимизировать транспортные расходы.

Рассмотрим решение транспортной задачи разными методами (минимального элемента, линейного программирования и в электронных таблицах Excel).

Условие задачи

Имеются две фабрики, выпускающие холодильники, которые поставляют их в три магазина. Первая фабрика выпустила в

январе 11 холодильников, а вторая — 14. Первому магазину для продажи требуется 10 холодильников, второму —8, третьему — 7.

Стоимость перевозки холодильника с любой фабрики в любой магазин известна и линейна (указана в табл. 9.2, например, в долларах).

Таблица 9.2. **Стоимость перевозки холодильников в три магазина**

*S*1	*S*2	*S*3
8	6	10
9	5	7

Сколько холодильников нужно отправить с каждой фабрики в каждый магазин, чтобы общая стоимость всех перевозок была минимальной?

Ход и результаты работы

1. *Решение методом минимального элемента.*

Обозначим фабрику F_i, магазин — S_i.

Чтобы построить математическую модель задачи, построим следующую таблицу (рис. 9.10).

Нижние треугольники соответствуют неизвестному числу холодильников, которые нужно перевезти с данной фабрики в определенный магазин.

Решаем задачу методом минимального элемента. Найдем два решения (рис. 9.11).

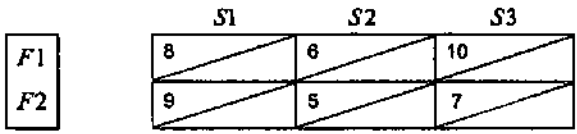


Рис. 9.10. Таблица для построения математической модели

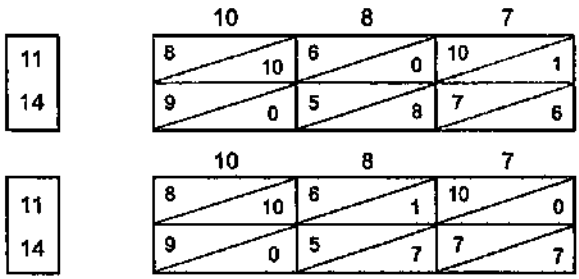


Рис. 9.11. Два решения задачи

Первый вариант решения дает следующие результаты:

Из F_1 в магазины отправлено $10 + 0 + 1 = 11$ холодильников, а из F_2 — $0 + 8 + 6 = 14$ холодильников. Стоимость перевозок $80 + 10 + 40 + 42 = 172$ долл.

Второй вариант решения дает следующие результаты:

Из F_2 в магазины отправлено $10 + 1 + 0 = 11$ холодильников, а из F_2 — $0 + 7 + 7 = 14$ холодильников. Стоимость перевозок $80 + 6 + 35 + 49 = 170$ долл.

Затраты на перевозки во втором варианте несколько меньше. Однако неясно, нет ли еще более оптимального варианта.

2. *Решение методом линейного программирования.*

Пусть x_{11} — число холодильников, отправленное из F_1 в S_1, x_{12} — число холодильников, отправленное из F_1 в S_2, x_{13} — число холодильников, отправленное из F_1 в S_3. В общем случае x_{ij} — число холодильников, отправленное из F_i в S_j. Тогда составим математическую модель задачи:

$x_{11} + x_{12} + x_{13} = 11;$

$x_{21} + x_{22} + x_{23} = 14.$

Магазинам при этом требуется:

$x_{11} + x_{21} = 10$ для $S1$;

$x_{12} + x_{22} = 8$ для $S2$;

$x_{13} + x_{23} = 7$ для $S3$.

Для каждого набора чисел x_{ij} общая стоимость перевозок равна:

$$8x_{11} + 6x_{12} + 10x_{13} + 9x_{21} + 5x_{22} + 7x_{23}.$$

Эти уравнения и есть ограничения нашей математической модели.

Введем еще ограничение неотрицательности $x_{ij} \geq 0$.

Математическая модель линейного программирования для решения нашей задачи будет иметь следующий вид:

$$8x_{11} + 6x_{12} + 10x_{13} + 9x_{21} + 5x_{22} + 7x_{23} \rightarrow \min;$$

$$x_{11} + x_{12} + x_{13} = 11;$$

$$x_{21} + x_{22} + x_{23} = 14;$$

$$x_{11} + x_{21} = 10;$$

$$x_{12} + x_{22} = 8;$$

$$x_{13} + x_{23} = 7.$$

Общее число переменных в нашей задаче $2 \cdot 3 = 6$ (число фабрик равно 2, число магазинов равно 3). Число уравнений в задаче равно сумме числа фабрик и магазинов $2 + 3 = 5$.

Решив систему уравнений с помощью несложных подстановок и арифметических операций, получаем следующие результаты:

$$x_{11} = 5;\ x_{12} = 5;\ x_{13} = 1;\ x_{21} = 5;\ x_{22} = 3;\ x_{23} = 6.$$

Подставив полученные значения в целевую функцию, получим:

$$8 * 5 + 6 * 5 + 10 * 13 + 9 * 5 + 5 * 3 + 7 * 6 = 182 \text{ долл.}$$

Стоимость перевозок выше, чем в первом и втором вариантах, но более равномерно распределены поставки холодильников в магазины.

3. *Решение в электронных таблицах Excel* (рис. 9.12, 9.13).

В строках 9—11 таблицы на рис. 9.13 представлены исходные данные, в строках 7—8 —данные из расчета на 1 холодильник, в строке 16 — результаты расчета стоимости перевозки одного холодильника по каждому пункту доставки. Например, для ячейки C16 это рассчитывалось по формуле C16=C7*C11+C8*C12. В ячейке B16 по формуле =СУММ(C16:E16) вычислялась общая стоимость перевозок одного холодильника до магазинов.

Оптимизация перевозок

	Потребители	Магазин №1	Магазин №2	Магазин №3
Поставщики				
Фабрика №1	3	1	1	1
Фабрика №2	3	1	1	1
	Факт	2	2	2
	Запросы	10	8	7
Фабрика №1	11	8	6	10
Фабрика №2	14	9	5	7
Всего	45	17	11	17

Рис. 9.12. Исходные данные

Далее в меню **Сервис** выбираем инструмент **Поиск решения**.

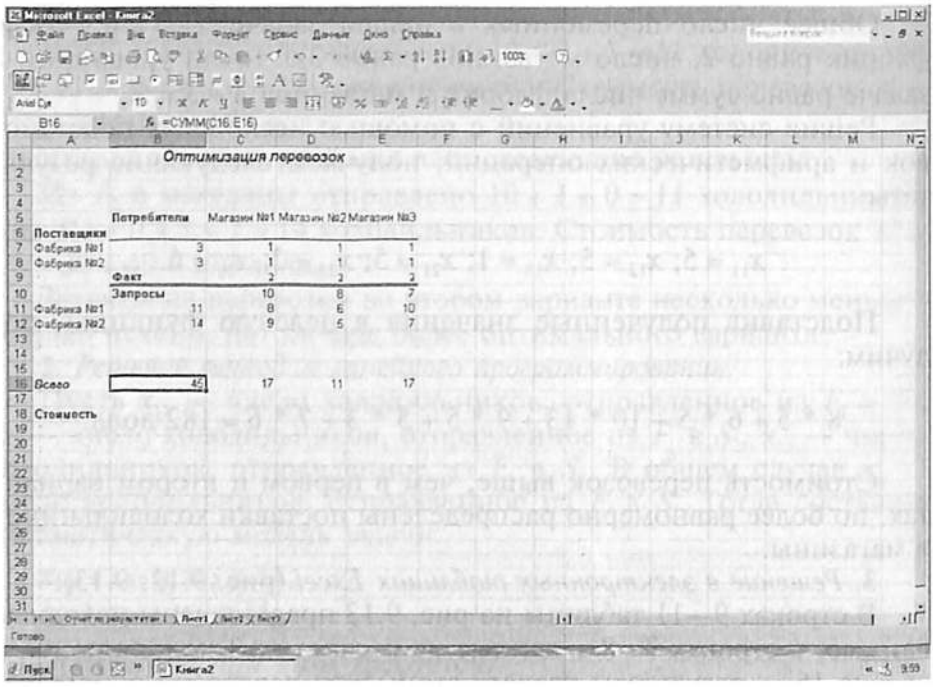


Рис. 9.13. Расчет стоимости перевозки одного холодильника

Поскольку в качестве критерия оптимизации выбрана минимизация стоимости перевозок, в поле **Установить целевую ячейку** следует ввести ссылку на ячейку $B$16, содержащую формулу расчета общей стоимости перевозок. Далее следует минимизировать значение этой ячейки путем изменения значений влияющих ячеек (изменяемых). Это ячейки, которые предназначены для хранения значений искомых неизвестных. Переключатель следует установить в положение минимального значения. В поле **Изменяя ячейки** вводим ссылки на ячейки с C7 по E8, т. е. будет изменяться стоимость груза, перевезенного по конкретному маршруту.

Далее надо наложить некоторые ограничения для поиска решения.

В группе полей **Ограничения** нажмите кнопку **Добавить**. Появится диалоговое окно **Добавление ограничения**. Следует ввести левую часть ограничения в левое поле, выбрать знак условия, накладываемого на значение, и ввести правую часть ограничения. Можно не вводить ссылки на ячейки, а выделить мышью

эти ячейки. После ввода одного ограничения следует нажать кнопку **Добавить** и ввести следующее. По окончании ввода всех ограничений (рис. 9.14) нажмите на кнопку **ОК**. В диалоговом окне появятся строки введенных ограничений. Первое условие — $B$7: $B$8 <=$B$11:$B$12. Оно означает, что значение в ячейке B7 должно быть меньше или равно значению в B11, в B8 — меньше или равно, чем в B12.

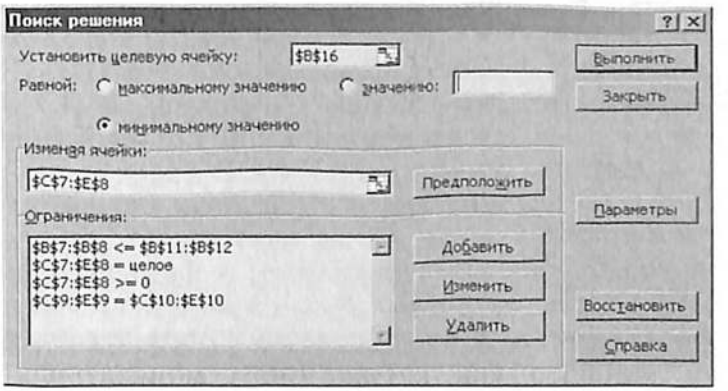


Рис. 9.14. Установка ограничений

Второе условие — $C$7:$C$8>целое. Это значит, что значения в этих ячейках должны быть целыми числами. Третье условие — $C$7:$C$8>=0. Оно означает, что объем перевозок не может быть отрицательным. Грузопоток имеет только одно направление — от фабрик к пунктам доставки. Четвертое условие — $C$9:$E$9>=$C$10:$E$10. Оно означает, что значения в ячейках девятой строки должны быть равны значениям в ячейках десятой строки, т. е. запросы пунктов доставки должны быть выполнены полностью. Введенные условия должны позволить найти наиболее оптимальный вариант решения задачи.

Далее следует нажать кнопку **Выполнить** для решения задачи.

После нахождения решения появляется диалоговое окно **Результаты поиска решения**.

Нажав кнопку **ОК**, занесем вариант решения на рабочий лист.

Из рис. 9.15 видно, что, при отправке 10 холодильников из первой фабрики в магазин № 1, а из второй фабрики — 8 холодильников в магазин № 2 и 7 холодильников в магазин № 3 будет затрачено 169 долл.

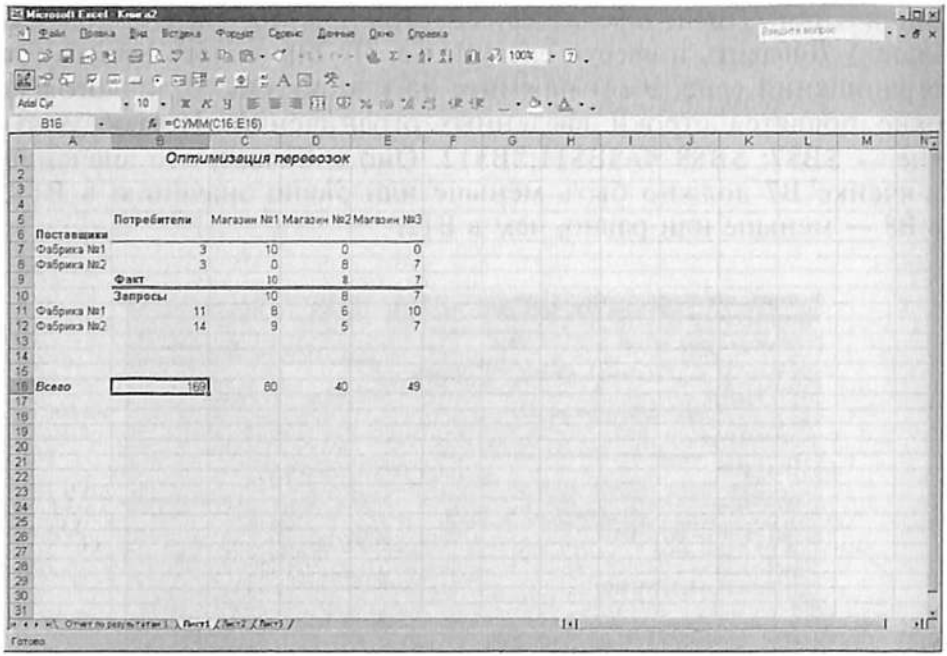


Рис. 9.15. Результат поиска решения

Сравнив результаты решения задачи тремя методами (первый метод дал затраты 170—172 долл., второй — 182 долл.), видим, что решение задачи в Excel дало оптимальный результат.

Контрольные вопросы

1. Каково назначение и состав математического обеспечения АИС?
2. Какие функции реализует система математического обеспечения АС?
3. Каковы этапы разработки МО?
4. Что такое математическое моделирование в АИС?
5. Что такое алгоритмическое моделирование в АИС?
6. Какова краткая характеристика метода исследования операций?
7. При решении каких задач используется метод линейного программирования и в чем заключается его сущность?
8. Что такое целевая функция и система ограничений?
9. В чем сущность метода минимального элемента?
10. Какие методы вы знаете для решения транспортной задачи?

9.3. Программное обеспечение

Программное обеспечение АИС — совокупность программ, обеспечивающих функционирование комплекса ее технических средств, реализацию целей и задач АИС.

ПО тесно связано с математическим обеспечением (МО), так как составляется на базе МО на основе алгоритмов.

ПО включает в себя ОС (операционные системы), ППП (пакеты прикладных программ) и системы программирования (СП).

ОС включает управляющую часть (УЧ) и обрабатывающую часть (ОЧ), ППП — программы общего и функционального назначения. СП осуществляют ведение информационной базы и организацию информационного процесса.

ОС является ядром ПО. УЧ обеспечивает решение задачи в требуемом режиме, ОЧ транслирует содержание задачи, записанной на некотором языке программирования, осуществляет редактирование программных модулей и генерирует необходимую конфигурацию вычислительной системы в целом. См. также разделы 2.7. и 2.8.

ОС может работать в следуюших режимах:

- индивидуальном (применяется для решения отдельных задач);
- пакетной обработки (потребитель не имеет доступ к ЭВМ, собранные им программы в пакет последовательно обрабатывает ЭВМ);
- многопрограммной работы (режим решения одновременно нескольких задач по различным программам);
- с распределением времени (наиболее развитая форма многопрограммной работы: абонентам одновременно предоставляют возможность общаться с ЭВМ и обращаться к общему информационному банку).

Основной принцип построения ОС заключается в выделении типовых процедур и оформлении их в виде стандартных блоков. Такой принцип называется модульным.

Программный модуль — программный блок, реализующий определенную функциональную возможность и рассчитанный на стандартные формы связи. Наиболее крупными программными блоками ОС являются супервизор и монитор.

Супервизор — совокупность программ, которые постоянно находятся в оперативной памяти и координируют поток задач через систему. Он распределяет ресурсы системы (время центрального процессора, оперативная память, устройства вво-

да-вывода и т. п.), планирует все операции при возникновении неисправностей, осуществляет обслуживание по таймеру.

Монитор — совокупность программ, которые обеспечивают управление решением задач на ЭВМ в различных режимах. Монитор воспринимает от работающих программ или от оператора управление (команды-директивы) и организует их выполнение.

Постоянно в ОП находится только часть монитора — резидент, остальная часть монитора хранится во внешней памяти и вызывается по мере необходимости.

В некоторых ЭВМ супервизор и монитор образуют программу диспетчер.

Особое внимание следует обратить на выбор сетевых ОС. Следует учитывать, насколько она способна взаимодействовать с другими ОС сети, как обеспечивает безопасность и защищенность данных, сколько пользователей может обслуживать, можно ли ее переносить на другую платформу и т. д.

Система программирования (*СП*) предназначена для автоматизации процесса программирования задач и содержит в своем составе удобную для работы программиста инструментальную оболочку, трансляторы алгоритмических языков высокого уровня и обслуживающие программы.

СП содержит средства для автоматизированной разработки и отладки программ, организации выполняемого процесса (ОВП) и ведения информационной базы (ВИБ).

СП могут быть одноязыковые (Visual Basic, Турбо Си, Турбо Паскаль и др.) и многоязыковые, т. е. когда отдельные части программных модулей написаны на разных языках (СП OS/360, СП UNIX и др.), а после компиляции они объединяются в исполняемые модули. Каждый язык программирования в большей степени пригоден для определенного класса задач (информационных, оптимизации и т. д.), поэтому система программирования содержит целый набор языков, которые в совокупности исчерпывают все типы задач, решаемых в АИС.

СП могут быть замкнутые и открытые, т. е. когда можно в систему добавлять язык программирования (ЯП) с транслятором. О ЯП см. раздел 3.3.

Пакеты прикладных программ (ППП), или приложения. ППП — совокупность программ, совместимых между собой и обеспечивающих решение задач из некоторой области знаний, называемой предметной областью пакета.

ППП — структурно-сложные системы программ, предназначенные для решения задач определенного класса. Проблемно-ориентированные системы предназначены для автоматизированного создания ПО. На их основе создаются ППП для вычислительного процесса и ведения информационной базы.

ППП могут быть программами общего назначения (ПОН) и программами функционального назначения (ПФН).

К ПОН можно отнести системы программирования на языках высокого уровня, СУБД, редакторы текстов, изображений, издательские системы и др. Первые реализуют типовые режимы работы вычислительной системы. К ПФН относят пакеты программ, предназначенные для решения задач в определенной предметной области, и др. Деление это достаточно условно.

Стандартная программа (прикладная) — общеупотребительная программа, построенная так, что ее можно включать в состав ППП для решения разных задач.

Библиотеки стандартных программ (БСП) формируются и содержатся на магнитных носителях (МН) под определенными именами (библиотеки статистической обработки данных, линейной алгебры, дифференцированного и интегрального исчисления, отыскания квадратного корня, нахождения экстремума и др.). Например, SSP — пакет научных прикладных программ — методы численного анализа и статистики.

При выборе ППП обычно следует учитывать следующие факторы:

1. Состав функций управления, реализуемых с помощью пакета; можно ли его принять полностью, или он должен быть доработан.

2. Возможность применения входных и выходных форм документов, регламентированных ППП.

3. Наличие исходных данных, регламентированных в ППП; возможность и трудоемкость их получения.

4. Возможность адаптации пакета и периодичность обработки данных различных пользователей.

5. Соответствие ППП необходимой структурной перестройке ОУ и степени оперативности реорганизации базы данных.

6. Надежность с точки зрения защиты данных, наличие средств обнаружения и локализации ошибок.

7. Наличие в пакете средств его развития и совершенствования.

8. Минимизацию или максимизацию конфигураций ЭВМ и периферийных устройств, которые предусматривает ППП.

9. Возможность использования различных носителей для формирования и хранения массивов.

10. Затраты на адаптацию ППП к другой конфигурации технических средств.

11. ОС для функционирования ППП.

12. Язык и транслятор, на котором написан ППП. Наличие необходимого транслятора у пользователя.

13. Состав стандартных вспомогательных программ, необходимых для применения пакета.

14. Наличие и комплектность документации для пользователя в соответствии с существующими нормативно-техническими документами.

15. Наличие документов по описанию применения ППП и обучению пользователей. Количество документации и полнота излагаемых вопросов с точки зрения привязки ППП к условиям пользователя.

Часто ПО АС предназначено для решения задач управления, учета, отчетности, планирования, управления производством, распределением ресурсов, кадрового учета, бухгалтерской финансовой деятельности и т. д. ППП получили широкое распространение как инструмент автоматизации проектирования АСУ, создания САПР АСУ (систем автоматизирования проектирования АСУ), проблемно-ориентированных систем. Для этой цели разработано большое количество общесистемных и функциональных ППП.

Например, ППП:

- АРИУС — реализует функции архитектурного проектирования АСУ;
- ISDOS, ADS.TAG — предназначены для формализации и автоматизации анализа системы;
- СОД, ТИС, ТЕКОД, ИНЕС — средства проектирования алгоритмов;
- СУБД: Clipper, Oracle и др.;
- редакторы текстов;
- реализации диалогов.

Следует обратить внимание на сетевые приложения. Это могут быть сетевые базы данных, почтовые системы, средства архивирования данных, системы автоматизации коллективной работы и т. д.

Таким образом, ПО можно разделить на внутреннее, обеспечивающее нормальную работу ЭВМ, и внешнее, позволяющее потребителю решать на ЭВМ необходимые задачи ввода, обработки, анализа и вывода информации наиболее простым и удобным образом.

Внутреннее ПО состоит из эксплуатационных (тестовых и диагностических) программ, проверяющих исправность оборудования ЭВМ, системы программирования и операционной системы (рис. 9.16).

Рис. 9.16. Классификация ПО

Внешнее ПО состоит из программ типовых процессов обработки данных в АИС (ввода, контроля, сортировки, корректировки, дублирования, поиска и вывода информации), программ решения конкретных задач и диспетчерской программы системы (рис. 9.16). Внешнее ПО решает конкретные задачи АС в соответствии с иерархическими уровнями системы управления.

Уровень 1: сбор данных о ходе производственного процесса от первичных датчиков и преобразователей и использование этих данных после обработки для прямого программного управления этими процессами.

Программы обеспечивают:

- опрос датчиков и преобразования по заданным алгоритмам;
- выработку управляющих воздействий на исполнительные органы.

Уровень 2: выбор методов обработки результатов измерений и вычислений необходимых параметров.

Уровень 3: оптимизация производственного процесса и адаптивное управление.

Уровень 4 (высший): информационное управление системы — административно-организационное управление от предприятия и выше).

Решаемые задачи планируются и управляются программами, написанными на основе методов исследования операций.

Для успешной реализации проекта должны быть построены полные и непротиворечивые модели архитектуры ПО, где отражаются иерархия подсистем и взаимодействие всех элементов системы. Наиболее верно структуру сложных систем отражает блочно-иерарахический подход к их исследованию и созданию ПО. При таком подходе сначала создают части объектов (блоки, модули), а затем выстраивают из них сам объект.

Проблемы создания ПО в сложных системах породили потребность в программно-технологических средствах специального класса — CASE-средствах.

Термин CASE (Computer Aided Software Engineering — разработка ПО с использованием компьютерной поддержки) подразумевал вначале автоматизацию разработки ПО, а теперь — процесс разработки сложных программных систем, т. е. программную инженерию.

Жизненный цикл ПО — одно из базовых понятий программной инженерии.

Жизненный цикл ПО (*ЖЦ ПО*) определяют как период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации. ЖЦ ПО регламентирован международным стандартом ISO/IEC 12207:1995 Information Technology Software Life Cycle Processes (Процессы жизненного цикла про-

граммного обеспечения). В этом стандарте Международной организации по стандартизации ПО (или программный продукт) определяется как набор компьютерных программ, процедур и, возможно, связанной с ним документации и данных. А процесс ЖЦ — совокупность взаимосвязанных действий, которые преобразуют входные данные в выходные.

В нашей стране создание ПО начиная с 1970-х годов регламентировалось стандартами ГОСТ ЕСПД — Единой системы программной документации — серия ГОСТ 19.xxx. Многие из этих стандартов устарели. В настоящее время процессы создания АС, в состав которых входит и ПО, регламентированы стандартами «Информационная технология. Комплекс стандартов на автоматизированные системы» (ГОСТ 34.601—90, 34.602—89, 34.603—92 и др.). Однако и в этих стандартах процессы создания сложных систем отражены недостаточно, поэтому для каждого проекта такой системы часто создают комплексы нормативных и методических документов, регламентирующих процессы создания конкретного прикладного ПО. Целесообразно использовать международные стандарты. Так, в указанном выше стандарте ISO/IEC 12207:1995 все процессы ЖЦ ПО разделены на три группы: основные, вспомогательные и организационные.

Реальный процесс разработки ПО, как правило, выполняется по одной из трех схем (моделей): спиральной, с промежуточным контролем, каскадной (рис. 9.17).

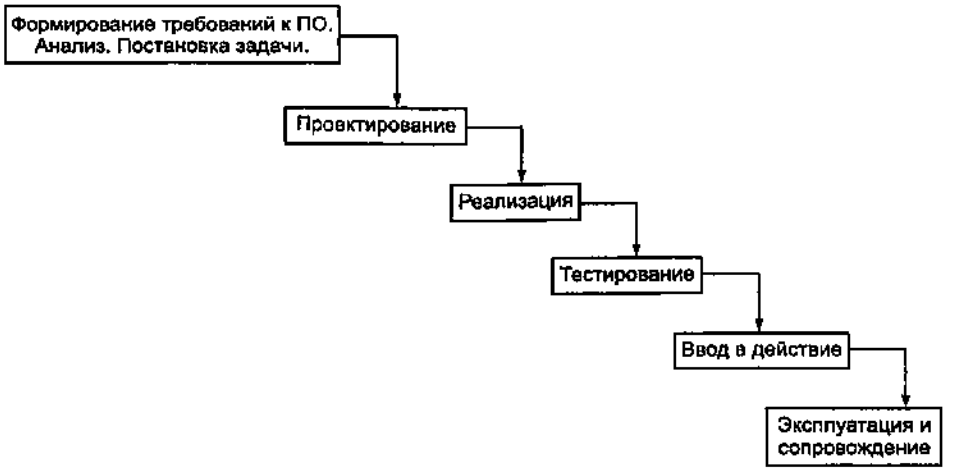


Рис. 9.17. Каскадная модель жизненного цикла ПО

На сегодняшний день существует два подхода в программной инженерии к разработке ПО систем:

- функционально-модульный, или структурный;
- объектно-ориентированный (объектная декомпозиция).

Принципиальное различие между ними обусловлено разными способами декомпозиции систем. В первом подходе выполняют разбиение на задачи подсистем по функциям. Во втором подходе структура системы описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами.

Технология программирования развивалась одновременно с развитием ЭВМ и языков программирования (см. раздел 3.2).

Технология прошла этап «стихийного» программирования, когда программы состояли из машинных кодов или ассемблеров и обрабатываемых данных.

Этап *структурного программирования* начался в 60—70-х годах прошлого столетия.

Структурный подход в сочетании с модульным программированием позволил разрабатывать надежные программы размером до ста тысяч операторов. Процедурные языки структурного программирования: PL/1, Алгол-68, Паскаль, Си. Языки, поддерживающие модульное структурное программирование: Паскаль, Си, Си++, Ада, Модула.

Для сложного программного обеспечения на этапе 80—90-х годов XX столетия стало применяться *объектно-ориентированное программирование*. Механизмы наследования, полиморфизма, композиции, наполнения позволили строить сложные объекты из простых. Были созданы среды, поддерживающие визуальное программирование: Delphi, C++ Builder, Visual C++ и др.

Компонентное программирование — создание ПО путем сбора объектов-компонентов (физически отдельно существующих частей ПО), взаимодействующих между собой через стандартизованные двоичные интерфейсы, в библиотеки или исполняемые файлы. О технологиях COM и CORBA см. раздел 3.2.

Современная тенденция заключается в стремлении приблизить язык программирования к человеческому языку, упростить его изучение и использование. О языках программирования см. раздел 3.3.

Использование машинных языков сохраняется еще в АСУТП, но при решении типовых задач обработки данных в АИС применяются языки высокого уровня и системы програм-

мирования. Автоматическое программирование — методы перевода с входного языка на машинный язык для работы по подготовке и программированию задач на ЭВМ — находит все большее распространение. Применяется мультипрограммирование, которое обеспечивает возможность использования для решения разных задач одних и тех же ресурсов ЭВМ, а также параллельной работы нескольких программ.

На сегодняшний день используют универсальные языки программирования, для которых характерны многоплатформенность, реализация всех основных структурных алгоритмических конструкций (условия, циклы), большие накопленные библиотеки подпрограмм и классов. Объектное представление программы использовано в новых версиях универсальных языков программирования: Object Pascal, Си++, Java и др.

Кроме универсальных, выделяют группы специализированных языков:

- баз данных (например, FoxPro, Oracle и др.);
- создания сетевых приложений (например, MySQL, SQL Server и др.);
- создания систем искусственного интеллекта (например, MYZIN и др.);
- пользователя (профессиональные среды пользователя).

Для эффективного пользования ПО необходимо разрабатывать грамотную, понятную пользователям программную документацию: руководство программиста, руководство пользователя, руководство системного программиста, пояснительные записки и др. Правила составления документов даны в указанных выше стандартах.

Контрольные вопросы

1. Каковы назначение и состав программного обеспечения АИС?
2. Какова структура программного обеспечения?
3. Что такое супервизор и монитор, каковы их функции?
4. Каковы назначение и функции системы программирования?
5. Что такое жизненный цикл ПО?
6. Каково назначение пакетов прикладных программ? Приведите примеры.
7. Что такое компонентное программирование?

9.4. Техническое обеспечение

Техническое обеспечение АС — совокупность средств реализации управляющих воздействий, средств получения, ввода, подготовки, преобразования, обработки, хранения, регистрации, вывода, отображения, использования и передачи информации и эксплуатационной документации (ГОСТ 2.601).

Состав КТС — это номенклатура и количество комплекса технических средств. В состав комплекса технических средств АИС входят:

1) средства подготовки и регистрации информации (СПР);
2) средства сбора и передачи информации (ССП);
3) средства хранения и обработки информации (СХО);
4) средства вывода и воспроизведения информации (СВВ).

Структура КТС — пространственное размещение ТС и система информационной связи между ТС и персоналом. В соответствии с ГОСТ 34.201—89 должна быть разработана структурная схема комплекса ТС АИС, составлен перечень и дано описание технических средств, составлены ведомость и спецификация оборудования и материалов, схема соединения внешних проводок, таблица соединений и подключений оборудования. Должна быть составлена инструкция по эксплуатации КТС.

Кроме КТС, к техническому обеспечению (ТО) относятся:

- ММ — методические материалы, включающие методику выбора КТС, библиотеки типовых программных решений для функционирования КТС, методику оценки показаний качества функционирования КТС;
- П — персонал по разработке, внедрению и эксплуатации ТС, включающий персонал: по ВТ, по периферийным средствам, по системам телеобработки данных, по средствам оргтехники и обслуживающий персонал.

КТС является одной из основных составных частей АИС, ее материально-технической базой, на которой реализуются все задачи системы. С помощью ТС реализуются функции автоматизированного сбора информации от первоисточников, ее анализ и представление, хранение, обработка, отображение, передача.

Для осуществления основных функций технические средства должны отвечать следующим требованиям:

- быть информационно совместимыми между собой и обслуживающим персоналом, что обеспечивается совпадением

форм представления информации, видов машинных носителей, языков, кодов, ввода данных в ТС;

- структура КТС должна соответствовать структуре управления объектом, обеспечивая автоматизированное управление выполняемых функций, в том числе функций контроля;
- для обеспечения быстрого решения задач ТС должны быть качественной конструкции, иметь современный и удобный дизайн, дружественный интерфейс для работы пользователя;
- должен соблюдаться принцип экономичности выбора и использования ТС, т. е. минимум затрат на создание (приобретение) технических средств, их эксплуатацию и используемых площадей для размещения.

Пример ТС для системы дорожного движения приведен в РД 50-34.698—90 [1].

В описании КТС приводятся:

1) общие положения;
2) структура КТС (схема);
3) средства вычислительной техники;
4) аппаратура передачи данных;
5) план расположения;
6) необходимое программное обеспечение.

В [1] для автоматизированного производства (для ИАСУ) выделяют ТС пяти уровней:

0 — устройства с числовым программным управлением (УЧПУ), локальные системы управления (ЛСУ), программируемые контроллеры (ПК). На этом уровне обеспечивается локальное управление в реальном времени;

1 — АСУ ГПМ (гибкий производственный модуль), работа в реальном времени группы ТС;

2 — АСУ ГАУ (гибкий автоматизированный участок), ГАЛ (гибкая автоматизированная линия);

3 — АСУ ГАЦ (гибкий автоматизированный цех);

4 — АСУП, АСНИ, САПР — связь между ними обеспечивается вычислительной сетью.

Например, система «Экспресс» (подробнее см. раздел 9.6.1), предназначенная для автоматизации обслуживания пассажиров в масштабе всей нашей страны, имеет технические средства в рамках следующих структур (рис. 9.18).

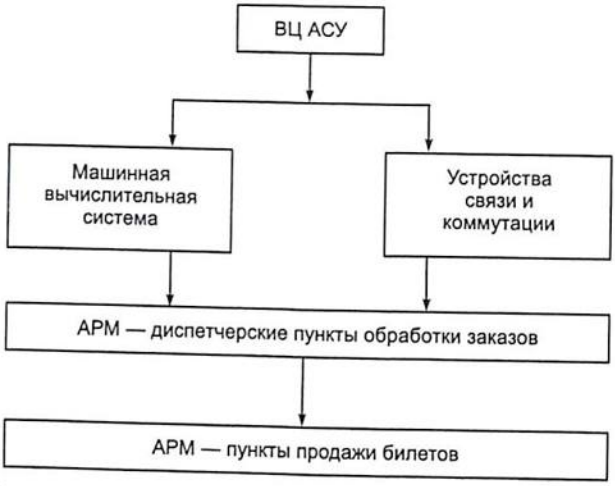


Рис. 9.18. Структуры системы «Экспресс», имеющие ТО

Управляет работой комплекса периферийных устройств АРМ устройство управления, построенное на базе микропроцессора. В АРМ в качестве устройств вывода служат:

- дисплей для визуального представления текстов оператора АРМ и ответов из ВЦ;
- билетопечатающее устройство;
- устройство вывода для изготовления статистической и отчетной документации;
- светооптическое табло для вывода данных о наличии мест в поездах.

Некоторые АРМ взаимодействуют между собой в режиме локальной сети, передавая данные с дисплея на дисплей, с МД одного АРМ на МД другого.

Массивы справочных данных хранятся на гибких магнитных дисках. Создан мощный справочно-информационный фонд, создается банк данных, содержащий информацию о номерах пассажирских поездов со всевозможными характеристиками, тарифно-справочные данные, таблицы кодирования станций и поездов, перечень пунктов продажи билетов и др.

Средства сбора и передачи информации (ССП) — различные датчики, ЭВМ, сетевое и телекоммуникационное оборудование, а также системы и средства связи общего назначения.

Датчики — устройства, преобразующие широкий круг информационных и технологических параметров в сигналы, которые могут быть обработаны в техническом устройстве (ЭВМ).

Датчики могут автоматически снимать с объектов такие параметры, как расходы количества вещества, состав газа, влажность и т. д. Но могут выдавать и экономическую информацию, например о загрузке оборудования, табельном учете, изготовлении и сдаче продукции, наличии документов и т. д.

Сбор информации может осуществляться в ручном, механизированном, автоматизированном и автоматическом режимах. Ручные и механизированные картотеки, автоматизированные хранилища документов и, наконец, сбор информации с помощью современных технических средств в настоящее время существуют, но все большее предпочтение отдается методам автоматизированного и автоматического сбора и передачи информации.

Современные ЭВМ содержат в своем составе большой и разнообразный набор устройств для запоминания, регистрации, отображения, ввода и вывода информации. Эти устройства называются периферийными. Огромное значение имеет не только совершенствование техники характеристик периферийных устройств, но и рациональная организация их работы. Скорость их работы ниже скорости работы процессора. Поэтому время, затрачиваемое на подготовку, ввод и вывод больших массивов данных, на порядки превышает время работы процессора. Задача — повысить скорость. Поэтому в современных ЭВМ введены унифицированные устройства управления вводом-выводом, обеспечивающие обмен информацией между ОП машины и внешними устройствами для осуществления их работы. Эти устройства называются каналами связи. Каждый канал — независимое устройство, подключаемое к процессору для управления обменом между ОЗУ и внешними устройствами. По способности одновременно обслуживать несколько периферийных устройств различают селективные и мультиплексные каналы.

Наиболее эффективно работают каналы, выполненные в виде небольших процессоров, осуществляющих операции обмена, контроля за правильной передачей информации, редактирование и анализ текста, распределение внешней памяти и различные операции по сортировке и группировке массивов. Широкие функции выполняют специальные процессоры управления обменом в устройствах связи с абонентами. Эти процессоры вводят информацию с телексов и выводят ее на телексы, факсы, осуществляют запись информации на внешний магнитный носитель, редактируют обмен информацией.

Устройства передачи данных — совокупность ТС и магнитных накопителей (МН), предназначенная для обмена информацией между ее источниками, потребителями и объектами управления. Обмен информацией происходит по каналам связи. Канал связи — совокупность ТС и физическая среда, предназначенная для передачи сигнала. Физическая среда, по которой распространяется сигнал, называется линией.

Системы связи: локальная, интегральная, территориальная, общегосударственная.

Наиболее распространенный режим связи абонентов — коммутация сообщений, когда тракт передачи информации организуется поэтапной передачей сообщений через центры коммутации по мере освобождения каналов данного направления.

Компьютерные сети классифицируют как локальные, глобальные, корпоративные, региональные (в том числе городские) — см. раздел 7.1.

Средства подготовки и регистрации информации (СПР). Самую большую сложность представляет перевод традиционных документов в электронный вид для последующей их обработки в ЭВМ. В настоящее время для этих целей используют клавиатуру ЭВМ, разнообразные сканеры, цифровые фотоаппараты, видеокамеры, мобильные телефоны, микрофоны.

Клавиатура служит для ввода символьных данных и управления работой ПК. Почти все клавиши работают по программе. Однако ввод содержимого документов вручную с клавиатуры для подготовки и регистрации их в ЭВМ — достаточно трудоемкий и утомительный процесс, поэтому все большее применение находят сканеры и другие технические средства, автоматизирующие его.

Сканер называют «глазами компьютера». *Сканер* — устройство, в котором с помощью лазерного луча считывается (точка за точкой переводится в цифровой вид) графическая информация (в том числе текст), которая воспринимается компьютером. Созданная в памяти компьютера совокупность точек является графической копией исходного документа. Затем путем распознавания с помощью специальных программ оптического распознавания символов (OCR) или шрифтового преобразования (например, FineReader российской фирмы ABBYY и CuneiForm российской фирмы Cognitive Technologies) графическое изображение текста можно перевести на естественный язык.

Основные характеристики сканера: оптическое и физическое разрешение, глубина цвета, скорость сканирования.

Разрешение — главная характеристика; чем оно выше, тем лучше. Сканирование выполняется по горизонтальным строкам, которые формируются точками. Чем больше светочувствительных элементов и чем лучше их качество, тем больше точек строки сканер воспринимает и тем выше разрешение. Единицей измерения разрешения является количество точек на дюйм (dpi). Современные сканеры имеют разрешение от 600 до 1200 dpi. Но это оптическое разрешение.

Физическое разрешение сканера зависит от шага перемещения каретки сканера вдоль изображения во время считывания. Это разрешение обеспечить проще, и производители часто рекламируют его.

Глубина цвета исчисляется в битах. Цветное машинное изображение строится из цветовых пятен красного, зеленого и синего цветов. Общее количество передаваемых цветов — около 16 млн. Для передачи каждого цвета требуется 8 бит, для трех — 24 бит. Современные сканеры имеют 48 бит, что позволяет лучше кодировать цвета и убирать искажения.

Скорость сканирования зависит от разрешения. Чем выше разрешение, тем медленнее сканирование. Например, для ввода страницы формата A4 черно-белого текста с разрешением 200 dpi скорость сканирования будет 2—40 с. Скорость сканирования зависит также от порта подключения к компьютеру. Самый быстрый, универсальный — USB, самый медленный, принтерный — LPT.

Ручные сканеры — самый простой вид сканеров, дающий наименее качественное изображение. Не имеют движущихся частей, и сканирование происходит путем перемещения сканера по документу. Недостатком является узкая полоса сканирования. Их используют для сканирования текстов, а для сканирования иллюстраций они непригодны.

Листовые (*протяжные*) *сканеры* — могут сканировать сразу весь лист целиком за один проход. Блок сканирования у таких сканеров неподвижен, а бумага проходит через сканер при помощи специальных валиков. Листовые сканеры гарантируют хорошее качество, но они предназначены для сканирования отдельных листов. Перевести при помощи этого сканера в электронную форму страницу книги или разворот журнала практически невозможно.

Планшетные сканеры обеспечивают наилучшее качество и максимальное удобство при работе с бумажными документами. Под крышкой планшетного сканера располагается прозрачное основание, на которое укладывается документ. Блок сканирования перемещается вдоль документа внутри корпуса сканера. Сегодня планшетные сканеры наиболее популярны, так как позволяют сканировать не только отдельные листы, но и переплетенные книги, журналы и т. п.

Разновидностью планшетного сканера является *сканер с прозрачным корпусом*, который может быть установлен вертикально, горизонтально и под углом. В этом сканере благодаря ударопрочному поликарбонатному стеклу корпуса хорошо видно, как заложен оригинал, что позволяет избежать его косой закладки.

Барабанные сканеры обеспечивают самое лучшее разрешение, но они предназначены не для сканирования бумажных форм, а для прозрачных материалов. В сканерах этого типа сканирующая головка установлена неподвижно, а цилиндр вращается с большой скоростью, сканирует построчно.

Сканеры форм предназначены для сканирования стандартных бланков. Этот вид сканера является подвидом листового сканера (используется, например, на выборах — для бюллетеней).

Штрих-сканеры— это разновидность ручного сканера; используется для считывания штрих-кодов (например, в магазине).

В нашей стране наибольшее распространение получили сканеры фирм Hewlett-Packard, Mustek и Epson.

Связь сканера с ОС. Для подключения сканера к ПК используют различные методы. Существуют сканеры со специальной платой расширения (контроллер). Плату устанавливают в одно из расширений материнской платы ПК, а сканер подсоединяют к разъему платы. Некоторые сканеры подключают к свободному параллельному порту, в таком случае передача данных идет медленнее, но зато без дополнительных устройств. Аппаратные интерфейсы сканеров отличаются многообразием, поэтому были приняты меры для стандартизации программного интерфейса, которые обеспечивают связь сканера с ОС. Если сканер поддерживает протокол (обычно протокол TWAIN), то ОС способна обеспечить взаимодействие между сканером и программным приложением, предназначенным для работы с текстом (например, ОС Windows способна обеспечить такое взаимодействие).

После сканирования документа получают его графический образ, который нужно распознать, отредактировать и записать в

формате, необходимом для его дальнейшего использования. Для этих целей служат специальные программы, например программа *FineReader* фирмы ABBYY.

На рис. 9.19 показан общий вид программы, а на рис. 9.20 — первое диалоговое окно Мастера сканирования и распознавания.

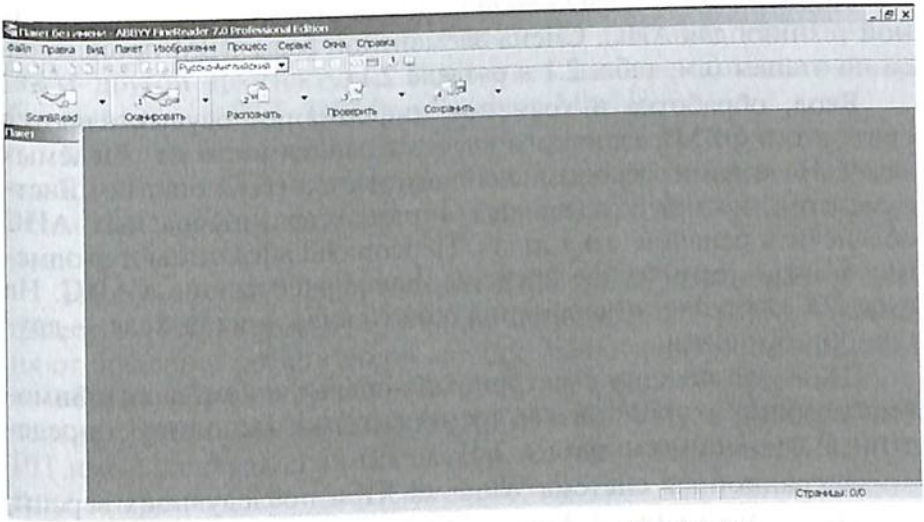


Рис. 9.19. Общий вид окна программы FineReader

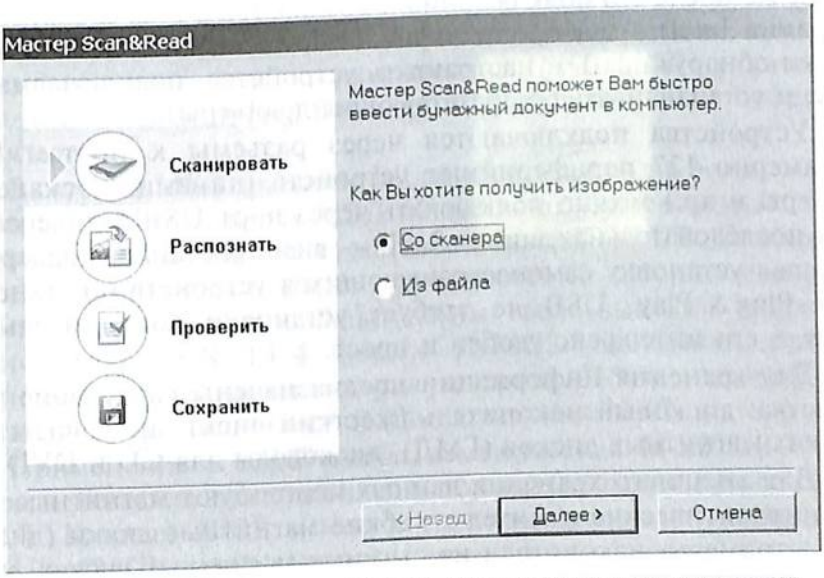


Рис. 9.20. Диалоговое окно Мастера сканирования и распознавания

Для ввода звуковой информации служит микрофон, который должен быть подключен к звуковой карте, находящейся в системном блоке компьютера.

Средства хранения и обработки информации (СХО). Компьютер — центральный компонент хранения и обработки данных.

Развитие ЭВМ давало толчок для создания более совершенной техники для АИС. Смена элементной базы ЭВМ происходила по этапам (см. табл. 2.1 в разделе 2.1).

Ввод, обработка и хранение информации осуществляются средствами ЭВМ различного класса в зависимости от решаемых задач. Но именно персональный компьютер (ПК) стал тем инструментом, который позволяет в рамках разнообразных АИС обеспечить решение этих задач. Персональный компьютер заменил многие технические средства, ранее работавшие в АИС. На вход ПК поступает информация одного вида, а на выходе — другая, обработанная.

ПК — это не один электронный аппарат, а комплекс взаимодействующих устройств, каждое из которых выполняет определенные функции (см. раздел 2.3).

Операционная система Windows XP и последующих версий, используя технологию Plug & Play (подключи и пользуйся), обеспечивает возможность работы с периферийным устройством сразу же после его подключения к компьютеру. Plug & Play (корпорации Intel) — набор спецификаций, позволяющий автоматически обнаруживать и настраивать устройства, подключаемые к ПК, и устанавливать соответствующие драйверы.

Устройства подключаются через разъемы к портам ПК. Примерно 127 периферийных устройств (динамики, сканеры, камеры и др.) можно подключать через порт USB (универсальная последовательная шина). USB — внешняя шина, поддерживающая установку самонастраивающихся устройств по технологии Plug & Play. USB не требует установки дополнительных плат, а его интерфейс удобен и прост.

Для хранения информации предназначены следующие устройства: дисковый накопитель (жесткий диск), дисководы для гибких магнитных дисков (ГМД), дисководы для CD и DVD.

Для внешнего хранения данных используют магнитные или магнитооптические носители: гибкие магнитные диски (дискеты), стримеры, накопители на съемных магнитных дисках, компакт-диски, флэш-диски и др. (см. раздел 2.6).

Основу современных АИС составляют базы данных (знаний), в которых накапливаются большие объемы информации, выполняются их обработка и хранение (см. главу 6).

Существующие стандартные СУБД для универсальных ПК уже не очень эффективно управляют этой работой. Считается, что необходимо создать специализированные АИС, которые бы реализовали только функции СУБД. Но для этих целей нужны ЭВМ другой архитектуры. Появились даже прототипы машин баз данных — МБД (DBM — Database Machine).

МБД предназначены в основном для работы с реляционными БД и используются в промышленности и коммерческих проектах. Выделяют следующие виды: многопроцессорные неоднородные и сетевые МБД.

Например, МН Delta состоит из четырех реляционных процессоров (РП), которые с высокой производительностью выполняют операции реляционной алгебры над отношениями большого объема. При этом каждый процессор может выполнять отдельную операцию независимо от других РП или параллельно. РП имеет центральный процессор, который выполняет логические операции. Два адаптера иерархической памяти и входной модуль помогают в работе с входными и выходными потоками данных. Процессор слияния (ПСЛ) и двенадцать процессоров сортировки (ПСО) осуществляют конвейерную сортировку отношений и слияние сортированных сегментов отношений. Для увеличения скорости ввода-вывода в МН МБД в качестве кэш-диска используется большая полупроводниковая буферная память.

В сетевых МБД хранение БД осуществляется на большом количестве НМД. Сетевые МБД объединяют универсальные микропроцессоры и устройства массовой памяти (УМП). Появились серийные однокристальные транспьютеры, которые содержат процессор, память и каналы (порты ввода-вывода), например транспьютер IMS T414 фирмы INMOS. Основной элемент транспьютеров — транспьютерная матрица. Появление этой техники привело к широкому созданию сетевых МБД.

Применение в АИС нашла коммерческая МБД DBC 1012 фирмы Teradata, включающая 8 обрабатывающих процессоров ПМД. Каждый ПМД имеет магнитные дисковые накопители (НМД). ПМД подключаются к сети, в узлы которой встроены высокоскоростные процессоры и программируемые управляю-

щие логические матрицы. К сети подключены коммуникационные процессоры для взаимодействия с главной ЭВМ.

Развивается также направление по созданию недорогих коммерческих устройств на серийных процессорных элементах с шинным интерфейсом, которые используются МБД (например, МБД IDM 500 фирмы Britton Lee).

Средства вывода и воспроизведения информации (СВВ). Эти средства описаны в разделе 2.5. Дисплей — необходимое устройство визуального вывода и воспроизведения информации на экране. Принтер — устройство вывода информации на печать. Акустические системы — устройства вывода звуковой информации.

Индикаторы — приборы (устройства), отображающие ход какого-либо процесса, полученные результаты, состояние объекта наблюдения и тому подобную информацию в простой и доступной человеку форме. Индикаторами могут служить лампа, звонок, стрелочный или цифровой прибор, табло.

Могут использоваться и другие устройства вывода, например большие экраны, плоттеры и др.

Выбор технических средств для решения конкретных задач. Рациональная структура КТС должна обеспечивать реализацию функций системы с минимальными затратами труда, материальных и денежных ресурсов. На стадии разработки технического задания на АИС проводят предварительный выбор ТС, а затем окончательный. При предварительном выборе определяют предстоящие затраты на технику и ее обслуживание и экономическую эффективность их работы. На основе анализа результатов обследования производственной, организационно-функциональной структуры организации (предприятия) и информационных потоков создают информационную модель, которая учитывает периодичность решения задач, обобщенные характеристики алгоритмов, условия работы системы, системные требования и ограничения, параметры ТС. На этой базе формируют разные варианты КТС. Выбирают тот вариант, который удовлетворяет наилучшему значению установленного критерия.

При выборе КТС необходимо учитывать:

- временные ограничения обработки информации;
- площади для размещения устройств;
- места возможного размещения устройств;
- характеристики оборудования;
- производственную структуру и технологию;
- тип деятельности организации (предприятия);

- взаимосвязь устройств, а в ряде случаев взаимоисключаемость;
- срок окупаемости и коэффициент эффективности затрат.

При окончательном выборе КТС уточняют исходные данные и получают объемные характеристики информационных массивов, что позволяет более точно определить необходимое количество средств обработки, преобразования и хранения информации. После выбора окончательного варианта составляется рабочая документация на КТС — схемы и чертежи размещения устройств, рабочий проект, монтажные схемы и др.

Критерием оценки (Z) вариантов обычно является минимум приведенных затрат на приобретение и эксплуатацию устройств. В качестве ограничений (требований) принимают время и достоверность преобразования информации. Основная задача выбора КТС заключается в том, чтобы найти множество устройств, минимизирующих функционал вида

$$Z = \text{Ен}\sum_{i=1}^{m}\sum_{k=1}^{n} q_{ik}\, a_{ik} + \sum_{i=1}^{m}\sum_{k=1}^{n} c_{ik} \rightarrow \min \text{ (целевая функция)}$$

при соблюдении ограничений

$$\sum_{k=1}^{n} T_{kj} \le \sum_{k=1}^{n} t_{dk} \le T_{dj};$$

$$1 - \sum_{k=1}^{n} (1 - P_k) \le P_{dj},$$

где Ен — нормативный коэффициент экономической эффективности капитальных вложений;

q_{ik} — количество устройств i-го типа, используемых на k-й фазе ($i = 1, 2, ..., m$; $k = 1, 2, ..., n$);

ik — стоимость устройства i-го типа, применяемого на k-й фазе;

c_{ik} — эксплуатационные затраты устройства i-го типа на k-й фазе;

T_{kj} — время преобразования информации на k-й фазе j-й задачи;

t_{dk} — допустимое время преобразования информации на k-й фазе;

T_{dj} — допустимое время преобразования информации по j-й задаче;

P_k — вероятность необнаруженных ошибок в процессе преобразования информации на k-й фазе;

P_{dj} — допустимая вероятность необнаруженных ошибок по j-й задаче.

При сравнении вариантов определяют для всех ТС функции ограничения выбора, стоимость и эксплуатационные расходы. Кроме того, для средств сбора и регистрации определяют места их установки, время регистрации. Для средств передачи информации — типы каналов связи, расстояния от пунктов сбора до пунктов обработки информации, время передачи. Для средств обработки — требования в зависимости от решаемых задач, характеристики устройств, время решения задач. Для средств выдачи и отображения информации — массивы информации, подлежащие хранению, пункты потребления и вид информации, время выдачи.

Для окончательного выбора варианта ТС принимают во внимание сроки решения каждой задачи и всей совокупности задач и осуществляют оптимальную компоновку ТС для всей технологической цепочки решения задач.

Срок окупаемости (T) и коэффициент эффективности единовременных затрат (E) на создание автоматизированной системы определяется по следующим формулам:

$$T = \frac{\text{Кд} + \text{Кн}}{\text{Э}}; \quad E = \frac{\text{Э}}{\text{Кд} + \text{Кн}},$$

где Кд — капитальные затраты на техническое оснащение (тыс. руб.) (КТС, строительство, реконструкция зданий);

Кн — предпроизводственные затраты на разработку и внедрение проекта (тыс. руб.);

Э — общий годовой экономический эффект от внедрения системы.

$$\text{Э} = [A_1 - (C_1 + \text{Ен}K_1)] - [A_0 - (C_0 + \text{Ен}K_0)],$$

где A_0 — стоимость годового объема товарной продукции до внедрения (тыс. руб.);

A_1 — стоимость годового объема товарной продукции после внедрения (тыс. руб.);

C_0 — себестоимость годового объема товарной продукции до внедрения (тыс. руб.);

C_1 — себестоимость годового объема товарной продукции после внедрения (тыс. руб.);

Ен — нормативный коэффициент экономической эффективности капитальных вложений;

K_0 — среднегодовая стоимость производственных фондов до внедрения системы (тыс. руб.);

K_1 — среднегодовая стоимость производственных фондов после внедрения системы (тыс. руб.).

В современных условиях особую актуальность приобретает проблема взаимодействия человека-оператора с техническими средствами АИС. Поэтому важное значение имеет унификация и стандартизация, использование типовых решений и модульного принципа проектирования систем отображения и обработки информации.

Контрольные вопросы

1. Каковы назначение и состав технического обеспечения АИС?
2. Какова структура комплекса технических средств (КТС) АИС?
3. Каковы требования, предъявляемые к КТС?
4. Какие вы знаете средства сбора и передачи информации? Кратко охарактеризуйте их.
5. Каковы средства подготовки и регистрации информации? Кратко охарактеризуйте их.
6. Какие вы знаете средства хранения и обработки информации? Приведите краткую характеристику некоторых из них.
7. Каковы средства вывода и воспроизведения информации? Кратко охарактеризуйте их.
8. Что необходимо учитывать при выборе КТС?

9.5. Правовое, организационное, методическое и эргономическое обеспечение

ГОСТ 34.003—90 дает следующие определения.

Правовое обеспечение АС — «совокупность правовых норм, регламентирующих правовые отношения при функционирова-

нии АС и юридический статус результатов ее функционирования».

Организационное обеспечение АС — «совокупность документов, устанавливающих организационную структуру, права и обязанности пользователей и эксплуатационного персонала АС в условиях функционирования, проверки и обеспечения работоспособности АС».

Методическое обеспечение АС — «совокупность документов, описывающих технологию функционирования АС, методы выбора и применения пользователями технологических приемов для получения конкретных результатов при функционировании АС».

Эргономическое обеспечение АС — «совокупность реализованных решений в АС по согласованию психологических, психофизиологических, антропометрических, физиологических характеристик и возможностей пользователей АС с техническими характеристиками комплекса средств автоматизации АС и параметрами рабочей среды на рабочих местах персонала АС».

Правовое обеспечение АИС. Главной целью правового обеспечения является укрепление законности. В состав правового обеспечения входят законы, указы, постановления государственных органов власти, приказы, инструкции и другие нормативные документы министерств, ведомств, местных органов власти.

Специалисты и другие пользователи, работающие с системой, прежде всего, должны знать действующие в стране законы, регламентирующие области работ, с которыми они соприкасаются. Им должны быть хорошо известны основные положения таких законов Российской Федерации, как:

1. Закон «Об информации, информационных технологиях и о защите информации» от 27 июля 2006 г. № 149-ФЗ.

2. Закон «О правовой охране для электронных вычислительных машин и баз данных» от 23 сентября 1992 г. № 3523-1.

3. Закон «О стандартизации» от 10 июня 1993 г. № 5154-1 (последняя редакция от 25 июля 2002 г. № 116-ФЗ).

4. Закон «О сертификации продукции и услуг» от 27 апреля 1993 г. № 5151-1 (в редакциях от 27 декабря 1995 г. № 211-ФЗ; от 2 марта 1998 г. № 30-ФЗ; от 31 июля 1998 г. № 154-ФЗ, изменения от 10 января 2008 ч. 4 ГК РФ).

5. Закон «Об участии в международном информационном обмене» от 4 июля 1996 г. № 85-ФЗ (последняя редакция от 29 июня 2004 г.).

6. Закон «Об авторском праве и смежных правах» от 9 июля 1993 г. № 5351-1 (в редакции от 19 июля 1995 г. № 110-ФЗ). Вместо этого документа с 1 января 2008 г. действует ч. 4 Гражданского кодекса (ГК) РФ.

7. Закон «Об электронной цифровой подписи» от 10 января 2002 г. № 1-ФЗ.

В этих законах определены основные принципы разработки, создания, сертификации и лицензирования информационных систем и средств их обеспечения, правовой режим участия в международном информационном обмене посредством АИС и др.

В 1990-е годы издан сборник межгосударственных стандартов «Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы». Эти стандарты устанавливают виды, наименование, комплектность и обозначение документов, разрабатываемых на всех стадиях и этапах создания автоматизированных систем.

Так, требования к содержанию документов, разрабатываемых при создании АС, устанавливают:

- Руководящий документ по стандартизации РД 50-34.698—90;
- Единая система программной документации (ЕСПД);
- Единая система конструкторской документации (ЕСКД);
- Система проектной документации для строительства (СПДС);
- ГОСТ 34.602.

В нашей стране действует Единая система программной документации (ЕСПД), представляющая собой комплекс взаимоувязанных государственных стандартов в области программирования. Эти стандарты регламентируют все виды программ и программной документации, процессы их разработки, оформления и обращения (сопровождение, тиражирование и др.). Например, при создании ПО для АИС необходимо выполнять требования ЕСПД, которая содержит:

- основополагающие и организационно-методические стандарты;
- стандарты, определяющие форму и содержание программных документов, применяемых при обработке данных;
- стандарты, обеспечивающие автоматизацию разработки программных документов.

Для создания АИС должны быть изданы приказы руководящего органа о начале работ и организации информационной

службы, а также общеотраслевые нормативные акты, регламентирующие отношения разработчика и заказчика. При разработке системы необходимо выполнять требования государственных и отраслевых стандартов на разработку информационного и другого обеспечения, государственных стандартов на разработку проектной документации. Необходимо разработать нормативные акты на получение и использование комплекса технических средств, издать правовое положение службы АИС (АСУ).

Функционирование системы осуществляется на основе положения о службах, обеспечивающих работу системы. Для персонала издаются должностные инструкции. При работе следует выполнять требования нормативных актов о создании, использовании и защите информации, о регламентировании технологического процесса ее автоматизированной обработки, об использовании вычислительной техники.

Отдельные виды АИС должны функционировать только в соответствии с их правовым положением, включая нормативные акты, связанные с договорными отношениями разработчика и заказчика, и правовым регулированием отклонений от договора.

АИС (или ее элементы) должна пройти лицензирование (разрешение на использование и распространение), а ее продукция — сертификацию (подтверждение на соответствие установленным требованиям).

Организационное обеспечение АИС. Это совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации АИС. Организационное обеспечение реализует следующие функции:

- анализ действующей системы управления организацией, для которой создается АИС, выделение процессов, подлежащих автоматизации;
- подготовку задач, подлежащих автоматизации;
- разработку управленческих решений по составу и структуре организации, методологии решения задач для повышения эффективности системы управления.

Организационное обеспечение АИС базируется на системе организационно-распорядительной документации. Эти документы применяются при оформлении распорядительной и исполни-

тельной деятельности органов управления и подчиненных им подразделений. Эти документы различают по способу получения, содержанию и назначению, стабильности реквизитов.

Организационное обеспечение включает не только документы, но и те структурные подразделения, которые осуществляют управление и функционирование системы.

Непосредственно с АИС работают:

- специалист (администратор) или группа специалистов, которые несут ответственность за поддержку данных в БД, ее целостность, защиту данных от несанкционированного доступа, работу с пользователями системы;
- операторы подготовки данных, которые, руководствуясь специальными инструкциями и умея работать с интерфейсом АИС, вводят данные в БД;
- разработчики приложений — специалисты, которые обращаются к БД через СУБД для написания нужных программ на языках программирования;
- интерактивные пользователи АИС — лица, имеющие доступ на ввод, обработку и вывод данных в определенной области;
- конечные пользователи — лица, получающие информацию из БД по запросам для использования ее при решении необходимых задач.

Организационными элементами системы являются коллективы людей или отдельные исполнители. Связи между организационными элементами могут быть информационные, соподчинения и взаимодействия. Оптимальная организация этих специалистов в структурные подразделения определяет эффективность разработки и функционирования системы.

Руководящий документ по стандартизации РД 50-34.698—90 рекомендует при создании АС разрабатывать по организационному обеспечению документ «Описание организационной структуры». В нем должны быть отображены изменения в организационной структуре управления объектом, организация подразделений и реорганизация существующих подразделений.

Методическое обеспечение АС. Методическое обеспечение АС напрямую зависит от сферы деятельности (управление, исследование, проектирование и т. п.) и предметной области, для которой она создавалась или в которой функционирует. Прежде всего, это справочники, методические указания и другая норма-

тивно-методическая литература: инструкции, руководства и т. п. Это может быть конструкторско-технологическая документация, плановые, учетные и отчетные документы, картотеки нормативов и др.

Руководящий документ по стандартизации РД 50-34.698—90 рекомендует разрабатывать по методическому обеспечению следующие документы:

- «Методика (технология) автоматизированного проектирования»;
- «Технологическая инструкция»;
- «Руководство пользователя»;
- «Описание технологического процесса обработки данных».

В документе «Методика (технология) автоматизированного проектирования» указывают основные пути и направления решения задачи, ограничения на решение, критерии оценки результатов. Описывают математические методы проектирования, состав и назначение процедур, порядок их взаимодействия. Определяют состав и формирование массивов информации, перечень обозначений элементов и диапазон изменения их значений, критерии оценки исходных данных, методы и модели решения. На каждой проектной процедуре указывают состав входных нормативно-справочных данных, правила доступа к ним, порядок выполнения, состав и форму выходных данных. Приводят анализ полученного решения на соответствие заданным критериям.

В документе «Технологическая инструкция» приводят выполняемые операции. Указывают наименование технологической операции (операций), приводят сведения о порядке и правилах ее (их) выполнения, перечень персонала, осуществляющего технологический процесс обработки данных.

Документ «Руководство пользователя» содержит, как правило, следующие разделы:

- *«Введение»* (указывают область применения, краткое описание возможностей, уровень подготовки пользователей, перечень эксплуатационной документации);
- *«Назначение и условия применения»* (указывают виды деятельности, автоматизируемые функции, средства автоматизации и условия их применения, ПО, носители данных, БД, требования к подготовке персонала);
- *«Подготовка к работе»* (указывают содержание дистрибутивного носителя данных, порядок загрузки данных и программ, проверки работоспособности);

- *«Описание операций»* (указывают содержание выполняемых функций, задач, комплексов задач и процедур, описание операций по обработке данных для выполнения указанных задач, дают подробное описание каждой операции);
- *«Аварийные ситуации»* (указывают действия при сбоях системы, отказах технических средств, разрушении целостности БД, при несанкционированном доступе и др.);
- *«Рекомендации по освоению»* (дают описание контрольного примера, правила его запуска и выполнения, другие рекомендации по освоению и эксплуатации).

В документе «Описание технологического процесса обработки данных» описывают технологический процесс сбора и обработки данных на периферийных устройствах при децентрализованной обработке данных или в вычислительном центре. Рассматривают состав и последовательность выполнения работ по сбору, регистрации, подготовке, контролю, обработке, передаче и отображению информации. Приводят перечень документации на каждую операцию технологического процесса.

Эргономическое обеспечение АС. Эргономическое обеспечение АС учитывает психологические, психофизиологические, антропометрические, физиологические характеристики и возможности пользователей АС при выборе и разработке технических средств АИС и организации благоприятной рабочей среды на рабочих местах персонала.

В требования по эргономике и технической эстетике входят показатели, которые задают необходимое качество взаимодействия человека с техническим средством (машиной) и комфортность условий работы персонала.

Контрольные вопросы

1. Что такое правовое обеспечение АИС?
2. Какие законы обеспечивают правовой статус АИС?
3. Что такое организационное обеспечение АИС?
4. Что такое методическое обеспечение АИС?
5. Какие документы должны быть разработаны для методического обеспечения АИС?
6. Что такое эргономическое обеспечение АИС?

9.6. Функциональные подсистемы АИС

Организационные формы функциональной части АИС. Рассмотрим сначала основные понятия.

Функция автоматизированной системы — это совокупность определенных целенаправленных действий АС.

Задача АС представляет собой функцию в виде формализованной совокупности автоматических действий, приводящих к получению намеченного результата.

Подсистема — выделенное из системы подмножество взаимосвязанных элементов, объединенных некоторым целевым назначением.

Функциональная часть АС — см. раздел 8.2.

В разных организациях структура подразделений, ответственных за функционирование АИС, может быть разной.

Например, работу функциональных подсистем АИС могут осуществлять следующие подразделения:

- операционная группа, состоящая из обслуживающего персонала, специалиста по охране данных, системного аналитика и системного программиста, специалиста по телекоммуникациям и др. Это подразделение обеспечивает работу и поддержку аппаратных средств, прикладного программного обеспечения, телекоммуникационной связи;
- группа создания приложений, состоящая из системного аналитика и программистов, менеджеров проектов и др. Это подразделение обеспечивает создание новых ИС;
- центральная информационная группа, состоящая из специалистов и обслуживающего персонала, отвечающих за работу с пользователями.

В другом случае организационно и технологически функциональные подсистемы могут осуществлять следующие подразделения:

- служба документационного обеспечения, реализующая организацию и обеспечение делопроизводства, документооборот, документационное обеспечение подготовки управленческих решений и контроль их выполнения;
- информационная служба (это может быть отдел автоматизации, вычислительный центр и т. п.) реализует сбор документированной и недокументированной информации из внешних источников, ее анализ, информационное оповещение и удовлетворение информационных потребностей

различных управленческих и технологических подразделений;

- экспертно-аналитическая служба, состоящая из аналитиков (советников, консультантов), осуществляет анализ различных ситуаций, вырабатывает альтернативные варианты решений, выполняет прогноз последствий различных управленческих решений.

Важной функцией всех информационных подразделений является поддержание целостности и сохранения информации. Это осуществляется путем пересмотра, ревизии и отсеивания утратившей актуальность информации. Сохранность информации осуществляется на основе нормативно-инструктивных документов.

В том или ином варианте подсистемы должны реализовать функции АИС по сбору и комплектованию, хранению (поддержанию целостности, актуальности и сохранности), поиску, обработке и выдаче информации пользователям системы.

Например, в составе АИС выделяют три подсистемы: сбора информации, представления и обработки информации, выдачи информации.

Подсистема сбора информации. *Сбор информации* — процесс и организационный порядок целенаправленного получения источников информации. Он предполагает получение информации, ее оценку с точки зрения полезности использования в АИС. Эта подсистема является организационно-технологической.

Порядок сбора информации отображается на организационных схемах, которые являются организационной основой подсистемы сбора информации.

Поступающая информация должна пройти предварительную обработку, т. е. она должна быть отнесена к определенным рубрикам.

Из разных источников (в том числе от поставщиков) по каналам связи осуществляется поступление информации. Разными способами (с помощью техники или вручную) специалистами выполняется оценка информации с точки зрения отнесения ее к рассматриваемой предметной области.

Специалисты группы организации получения информации должны налаживать каналы связи с поставщиками информации, обслуживать каналы ее поступления.

Сбор и передача информации осуществляются в настоящее время в основном по компьютерным сетям. См. главу 7. Класси-

фикация сетей по признакам функциональности, целевого назначения, области использования представлена на рис. 9.21.

Рис. 9.21. Классификация компьютерных сетей по признакам функциональности, целевого назначения и области применения

Вычислительные сети производят обмен данными между пользователями сети для решения необходимых задач.

Информационные сети осуществляют информационное обслуживание по запросам пользователей.

Информационно-вычислительные сети выполняют функции двух предыдущих видов сетей.

Универсальные сети предоставляют услуги разнообразным пользователям.

Специализированные сети — сети управления производством и предприятиями, организациями, фирмами, компаниями.

По области использования в составе автоматизированных систем обработки информации (АСОИ) компьютерные сети находят широкое применение во всех сферах деятельности.

По высокоскоростным локальным сетям сбор и передача информации обеспечиваются различными услугами файловой службы, баз данных, печати, факсимильных сообщений, электронной почты и др.

Глобальные сети обеспечивают передачу файлов из публичных архивов удаленных серверов и почтовые услуги. Структура

Интернета построена по типу клиент/сервер, где клиенты-компьютеры получают информацию из Сети, а компьютеры-серверы в основном накапливают информацию и отдают ее клиентам.

В настоящее время широко используется гипертекстовая информационная служба WWW, которая применяет интранет-технологию (Intra — внутренний), задействующую службы глобальных сетей в локальных (внутренних).

С помощью этой технологии можно оперативно поместить необходимую информацию на компьютеры корпоративной сети (веб-серверы). Информацию можно быстро просмотреть с помощью специальных программ — веб-браузеров (стандартных навигаторов по Сети). Многие организации осуществляют сбор информации со страниц веб-серверов, где хранится огромное количество документов разных организаций. Эта технология улучшает процесс обмена информацией и взаимодействия между поставщиками и потребителями информации. Электронная почта снижает зачастую потребность в передаче информации по телефону или обычной почте. По сетевым каналам можно передавать не только компьютерные данные, но и голосовую и видеоинформацию.

При передаче информации по сетям большое внимание уделяется методам ее защиты: шифрованию данных, аутентификации пользователей, защитным барьерам.

В последнее время появились технологии, объединяющие все типы обмена данными в одной транспортной сети (например, ATM). Они используются для передачи данных в разных сетях: локальных, глобальных, телефонных, широковещательных и т. д.

Особенно следует выделить электронную почту — наиболее старую и распространенную службу обмена информацией. По электронной почте пользователи передают сообщения через модем или сетевое соединение с одного компьютера на другой.

Подсистема представления и обработки информации. Подсистема составляет ядро системы и является одним из наиболее сложных компонентов при ее разработке.

Представление и обработка информации — процессы и организационный порядок целенаправленного окончательного отбора, представления и обработки информации. Порядок отбора информации, ее оценка и фиксация предполагаются с точки зрения полезности использования в АИС. Выполняется комплектова-

ние — процесс сложения информации из множества частей в единое целое для доведения ее для дальнейшего использования.

Представление информации должно отражать структуры, картины и сведения о предметной области АИС.

Порядок отбора и фиксации информации отображается на организационных схемах. Информация должна быть классифицирована и в структурированной форме занесена в информационную базу.

Структурирование информации — процесс перевода информации (документированной и недокументированной) с формата и языка ее поступления в формат и язык представления данных в АИС. Информация должна быть представлена в виде унифицированных форм документов или поисковых образов документов.

Наполнение (комплектование) информационной базы АИС данными производится путем занесения (ввода) подготовленной информации в базу и установления взаимосвязей (в том числе и логических) новых данных с уже имеющимися.

Поиск и выдача данных — организационно-технологический порядок обработки данных для удовлетворения информационных потребностей пользователей АИС в различных сферах деятельности.

Информационным фондом подсистемы является *база данных* (*БД*). Для создания и рационального функционирования БД должна быть разработана или выбрана одна из стандартных СУБД, позволяющая создавать и хранить большие массивы данных и манипулировать ими. Совокупность БД, СУБД, системы унифицированной документации (формы входных документов, запросов) и технических средств образуют *банк данных* (*БнД*). В некоторых АИС информационным фондом является *база знаний* (*БЗ*).

В подсистеме должны быть обеспечены технологические процессы отбора, фиксации, накопления, обработки, поиска и вывода данных.

Специалисты группы отбора информации должны хорошо знать предметную область АИС, а также уметь квалифицированно отбирать и индексировать информацию. Специалисты группы фиксации информации должны выполнять ее структурирование. Специалисты группы ввода информации должны выполнять ее ввод в информационную базу. Специалисты группы накопления, обработки, поиска и вывода информации должны

осуществлять эти процессы. Группа состоит из наиболее квалифицированных операторов и программистов. Периодически они осуществляют плановый поиск и выдачу сведений для обеспечения производственной деятельности организации. Это могут быть статистические и сводные данные на разные периоды деятельности организации. Информационное обслуживание может осуществляться в виде периодического оповещения производственных структур, а также в виде выдачи информации по запросам. Информация может выдаваться по любой тематике, имеющейся в АИС. Обработка запросов и выдача по ним необходимых данных является важнейшей функцией АИС.

При создании и поддержании в актуальном состоянии банков данных администраторы АИС должны обеспечивать их целостность (неразрушение внутренних взаимосвязей информационных объектов), резервирование, защиту от несанкционированного доступа, возможность восстановления после разрушений или сбоев в работе.

Подсистема выдачи информации. Подсистема является нормативно-функциональной.

Выдача информации — процессы и организационный порядок определения пользователей (абонентов) системы и обеспечение их необходимой информацией. В подсистеме определяются режимы и формы выдачи информации из АИС. Ведутся учет и документирование выданной информации.

Поиск и выдача информации — установление специального технологического порядка удовлетворения информационных потребностей абонентов ИС и управленческой деятельности в технологических процессах.

Функциональные подсистемы АСУ. Из-за сложности задач управления современным предприятием возникает необходимость в функциональном разделении его на подсистемы. Процесс их формирования и установления между ними связей протекает в зависимости от возможности алгоритмизации процессов в системе.

Функциональная часть АСУ — комплекс административных, организационных и экономико-математических методов, обеспечивающих решение задач планирования, учета и анализа показателей для принятия управленческих решений в подсистемах. Подсистемы формируют, как правило, по фазам (планирование, учет, контроль, регулирование) или по функциям. Формирование подсистем по функциям определяет экономическую модель

предприятия (организации). Например, на предприятии это могут быть подсистемы управления:

- технической подготовкой производства (ТПП);
- технико-экономическим планированием (ТЭП);
- оперативным управлением основного производства (ОУОП);
- материально-техническим снабжением (МТС);
- сбытом и реализацией продукции (СРП);
- бухгалтерским учетом (БУ) и др.

Подсистема управления ТПП — комплекс работ по проектированию и освоению производства новых и совершенствованию выпускаемых изделий и технологических процессов их изготовления.

ТПП включает этапы:

- научно-исследовательских работ;
- конструкторской подготовки производства;
- технологической подготовки производства;
- опытного производства;
- инструментального производства.

Не все работы могут быть автоматизированы. Выбирать следует целесообразные функции, подлежащие автоматизации. Например, автоматизации подлежат организация фонда нормативно-справочной информации, разработка плана технической подготовки производства новых изделий, определение узлового и подетального состава изделий, определение материальных и трудовых затрат на производство изделий и т. д.

Подсистема управления ТЭП — предназначена для автоматизации наиболее трудоемких работ по расчету объемных, количественных и важнейших качественных показателей работы предприятия и его подразделений. В этой подсистеме выполняют разработку и обоснование перспективных и текущих (годовых) планов. Планы разрабатывают с выделением работ по годам. Годовой план имеет своей целью обеспечить взаимоувязку трудовых, материальных и финансовых показателей работы.

В подсистеме автоматизации подлежат основные решаемые задачи:

- расчет производственной мощности предприятия;
- расчет плана реализации продукции и прибыли;
- расчет амортизационных отчислений;
- расчет плана по труду и зарплате;
- составление отчетной документации о выполнении производственной программы;

- планирование себестоимости продукции (прямые расходы);
- расчет оптимальной производственной программы на год;
- разбиение годовой производственной программы по календарным отрезкам времени (квартал, месяц, декада) — объемно-календарное планирование.

Подсистема ОУОП — предназначена для разработки и установления для подразделений предприятия заданий по выполнению производственной программы на короткие отрезки времени (год, смена, час), учета, контроля и регулирования выполнения этих заданий.

В этой подсистеме проводятся оперативное планирование, оперативный учет и анализ хода производственного процесса, оперативное регулирование, обеспечивающее выработку управляющих воздействий на элементы производственного процесса с целью ликвидации возникающих отклонений. Учет и анализ должны быть четкими, точными, объективными и своевременными.

В подсистеме автоматизации подлежат задачи оперативного планирования:

- составление календарных планов по выпуску продукции в определенные сроки разными подразделениями;
- обеспечение максимально уплотненной загрузки оборудования, производственных площадей и рабочих, минимальной длительности производственного цикла и других условий для дальнейшего повышения производительности труда;
- обеспечение равномерного выпуска готовой продукции;
- обеспечение планирования между подразделениями.

В подсистеме автоматизации подлежат задачи оперативного учета:

- выполнение плана по изготовлению и сдаче продукции;
- наличие заготовок, полуфабрикатов, потерь от брака;
- процент выполнения плана по подразделениям.

Подсистема управления МТС — предназначена для своевременного и комплексного обеспечения подразделений и рабочих необходимыми материальными ресурсами.

В подсистеме подлежат решению задачи управления:

- формированием фондов на материальные ресурсы;
- заказами на материальные ресурсы;
- запасами материальных ресурсов;
- распределением материальных ресурсов внутри предприятия.

Для решения задачи управления формированием фондов на материальные ресурсы должны быть автоматизированы расчеты потребности в материалах и покупных изделиях и формирование заявок на фонды.

При решении задачи управления заказами на материальные ресурсы на базе специальных норм расхода должны быть автоматизированы расчеты определения потребности в материальных ресурсах на определенный период времени и налажены каналы связи с поставщиками.

Для решения задачи управления запасами материальных ресурсов должны быть автоматизированы расчеты:

- норм запаса материалов и покупных изделий (фаза планирования);
- средних значений запасов (фаза учета);
- отклонений величины запасов от нормативов (фаза контроля и анализа);
- корректировки заказов и заявок на фонды (фаза регулирования);
- учета выдачи материалов и покупных изделий подразделениям предприятия;
- контроля обеспечения материалами и покупными изделиями;
- перераспределения материальных ресурсов между подразделениями предприятия.

Подсистема управления СРП — предназначена для планирования, учета и управления поставками и реализацией готовой продукции предприятия. Управляющий орган этой подсистемы — финансово-статистический отдел (ФСО). Объектом управления являются потоки готовой продукции, отгружаемой со складов транспортной службой.

Основная задача этой подсистемы — организация и обеспечение поставками потребителей в соответствии с договорами, получение запланированной суммы реализации продукции. Эта подсистема сложнейшая, нелинейная и дискретная. Она испытывает очень много внешних воздействий.

В подсистеме решаются задачи:

- планирования поставок, отгрузки и реализации продукции;
- прогнозирования поставок, отгрузки и реализации продукции, а также штрафных санкций;
- оперативной оценки выполнения планов поставок и реализации продукции и выработки управляющих воздействий;

- учета и отчетности;
- формирования и ведения информационной базы подсистемы.

Для решения задач управления сбытом и реализацией продукции должны быть автоматизированы расчеты оптимизационных задач:

- распределения поставок и реализации по срокам и потребителям;
- комплектации и маршрутизации поставок;
- прогнозирования поставок и реализации продукции;
- выявления величин рассогласований между плановыми и фактическими заданиями;
- учета и статистической отчетности движения готовой продукции и учета портфеля заказов.

Автоматизируются задачи формирования информационных массивов подсистемы, вычисления нормативных параметров и др.

Подсистема управления БУ (бухгалтерского учета) — предназначена для анализа и контроля производственно-хозяйственной деятельности предприятия.

В подсистеме решаются задачи двух классов:

1) с преобладанием учетных операций, когда первичная документация группируется по источникам поступления средств и направлениям расходования;

2) с проведением сложных вычислительных операций, например расчеты с использованием классификаторов по материалам, труду и т. д. В результате таких расчетов получают информацию о расходовании поступающих средств.

В подсистеме автоматизируются задачи учета:

- основных и денежных средств;
- материалов и материальных затрат;
- готовой продукции;
- банковских и кредитных операций;
- труда и заработной платы;
- калькуляции себестоимости и др.

АСУ могут включать и другие подсистемы.

АСУ последнего поколения — это гибкие, адаптированные, интегрированные системы, включающие в себя элементы искусственного интеллекта. Посредством таких систем осуществляется переход к безбумажной и малолюдной технологии управления. В АСУ находят применение элементы самообучения и самонастройки на широкий класс объектов управления. Технической

базой такой АСУ служит супер-ЭВМ четвертого поколения, объединенная сетью с мини- и микро-ПЭВМ. Для развития ПО применяются языки высокого уровня и средства интеллектуализации.

9.6.1. Примеры функционирующих АИС

Автоматизированная информационная система «Реестр прав». Систему «Реестр прав» учреждение юстиции Новосибирской области разработало для себя и заинтересовано в ее поддержке независимо от возможных изменений законодательства и условий финансирования учреждений юстиции. АИС предназначена для комплексной автоматизации работы учреждения юстиции и его филиалов (от отдельного филиала, имеющего только один компьютер, до учреждения юстиции в целом). Предусматривается ведение Единого государственного реестра прав (ЕГРП) в электронном виде на основе создания системы сбора, обработки, накопления, хранения, поиска, распространения и предоставления информации о зарегистрированных на территории регистрационного округа правах на недвижимое имущество и сделках с ним в соответствии с действующим законодательством.

АИС «Реестр прав» представляет собой единый программный продукт и построена в двухзвенной архитектуре клиент/сервер на базе промышленной СУБД DB2 IBM 8.0.

Основные принципы, заложенные при разработке АИС «Реестр прав»:

- преимущество требований действующего законодательства перед техническими возможностями АИС «Реестр прав»;
- соответствие технологического процесса, заложенного в АИС «Реестр прав», реальной работе структурных подразделений учреждения юстиции;
- единообразие ведения ЕГРП, предоставления информации и организации документооборота в рамках одного регистрационного округа;
- гибкость АИС, возможность адаптации под изменения технологического цикла обработки дел правоустанавливающих документов, сложившихся в учреждении юстиции;
- соблюдение оптимального соотношения между свободой действий пользователя АИС и ограничениями, обеспечи-

вающими соблюдение действующего законодательства, безопасность информации, а также целостность базы данных;
- разграничение прав доступа и полномочий групп пользователей (отдельных пользователей) в отношении информации как на уровне меню пользователей, так и на уровне таблиц базы данных;
- возможность выполнять все предусмотренные технологическим циклом действия в рамках одной программы (без использования дополнительных программ и текстовых редакторов).

Основными объектами, используемыми в модели АИС, являются следующие:
- объекты недвижимого имущества — земельные участки, участки недр, обособленные водные объекты и все, что прочно связано с землей, т. е. объекты, перемещение которых без несоразмерного ущерба их назначению невозможно, в том числе леса, многолетние насаждения, здания, сооружения;
- пакеты входящих документов — пакеты, формируемые при приеме документов, основа АИС;
- субъекты прав — физические либо юридические лица, участники правоотношений с объектами недвижимости;
- правоустанавливающие документы и документы ЕГРП;
- регистрационные действия — процесс формирования регистрационных записей о государственной регистрации прав, сделок, ограничений, обременений по объекту недвижимого имущества на основании правоустанавливающих документов (рис. 9.22).

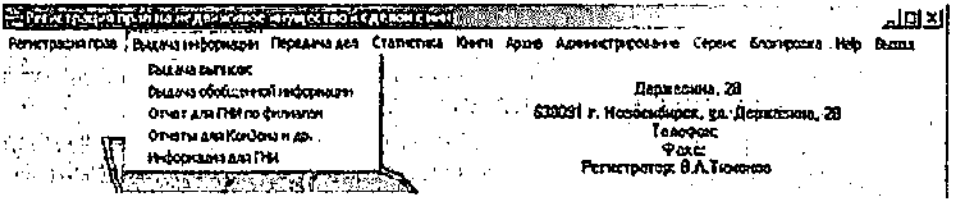


Рис. 9.22. Электронный бланк регистрации прав на недвижимость

Важной особенностью АИС «Реестр прав» является принцип многократного использования данных, единожды введенных в систему.

Состав АИС:

1. *Информационные ресурсы*:
   - база данных о зарегистрированных объектах недвижимости и сделках с ними на территории определенного региона;
   - банки данных, где информация распределена по группам, каждая из которых представляет специализированную информацию для сотрудников учреждения юстиции.

2. *Персонал, использующий АИС*: операторы, являющиеся сотрудниками учреждения юстиции; они вносят необходимые данные, а также составляют отчеты.

3. *Процедуры технологии*:
   - ввод и корректировка различных данных о физических и юридических лицах, а также их недвижимости и сделках;
   - просмотр необходимых данных по вышеперечисленным объектам;
   - создание и формирование отчетов по выбранным параметрам объектов.

4. *Интерфейс* — совокупность стандартных объектов Borland Delphi, позволяющая операторам использовать возможности АИС «Реестр прав» для решения поставленных задач.

5. *Комплекс технических средств и программного обеспечения*:
   - на базе каждого филиала учреждения юстиции формируется локальная вычислительная сеть (ЛВС), включающая сервер БД и клиентские рабочие места;
   - на базе головного учреждения юстиции тоже формируется ЛВС, включающая сервер БД и клиентские рабочие места, а также серверы и клиентские рабочие места удаленных городских отделов. На сервере головного учреждения юстиции формируется объединенная база данных по регистрационному округу;
   - ЛВС филиалов и головного учреждения могут иметь онлайновое соединение, в противном случае пополнение объединенной БД осуществляется данными, передаваемыми в управление юстиции по установленному регламенту (например, ежедневно). С филиалами, работающими в онлайновом режиме, возможен обмен между БД не только изменившейся, но и полной информацией за установленный период.

Система рассчитана на работу с базами данных промышленного уровня, имеет дружественный, интуитивно понятный интерфейс, поведение программы предсказуемо для пользователя. Программное обеспечение АИС может быть предоставлено другим учреждениям юстиции бесплатно независимо от количества рабочих мест. Структура БД открыта для специалистов информационно-вычислительных отделов учреждений и доступна для анализа. Предусмотрена возможность выгрузки изменений данных, имевших место на нижнем уровне, и их передачу (по каналам связи либо с использованием магнитных носителей) на вышестоящий уровень.

Структура АИС представлена на рис. 9.23.

Автоматизированная информационная система «Экспресс-3». Система разработана в рамках проводимой программы информатизации на железнодорожном транспорте. Она должна автоматизировать управление пассажирскими перевозками для обеспечения устойчивого функционирования российских железных дорог на рынке транспортных услуг. Автоматизация всех основных технологических процессов, включая управление багажными и грузобагажными перевозками, справочно-информационным обслуживанием пассажиров, управлением эксплуатацией и ремонтом парка пассажирских вагонов, оперативным планированием и прогнозированием, имеет важное экономическое значение.

В 1972 г. была создана первая опытная система электронного резервирования мест «Экспресс-1». Затем, в 1982 г., — типовая система для сети железных дорог «Экспресс-2». Накопленный опыт по их созданию и эксплуатации показал, что эти системы дают возможность автоматизировать все основные технологические процессы пассажирского хозяйства и создать на железных дорогах механизм оперативного управления пассажирскими перевозками. В составе систем создаются автоматизированные диспетчерские центры управления (АДЦУ) на каждой дороге и в министерстве.

Создание системы на базе современных ЭВМ позволит завершить полную автоматизацию всех технологических процессов пассажирского хозяйства и перевозок, а также оперативно управлять использованием подвижного пассажирского состава с гибким изменением тарифов, подняв «населенность» на 10—15 % на основе автоматизации маркетинговых исследований и культуру обслуживания пассажиров.

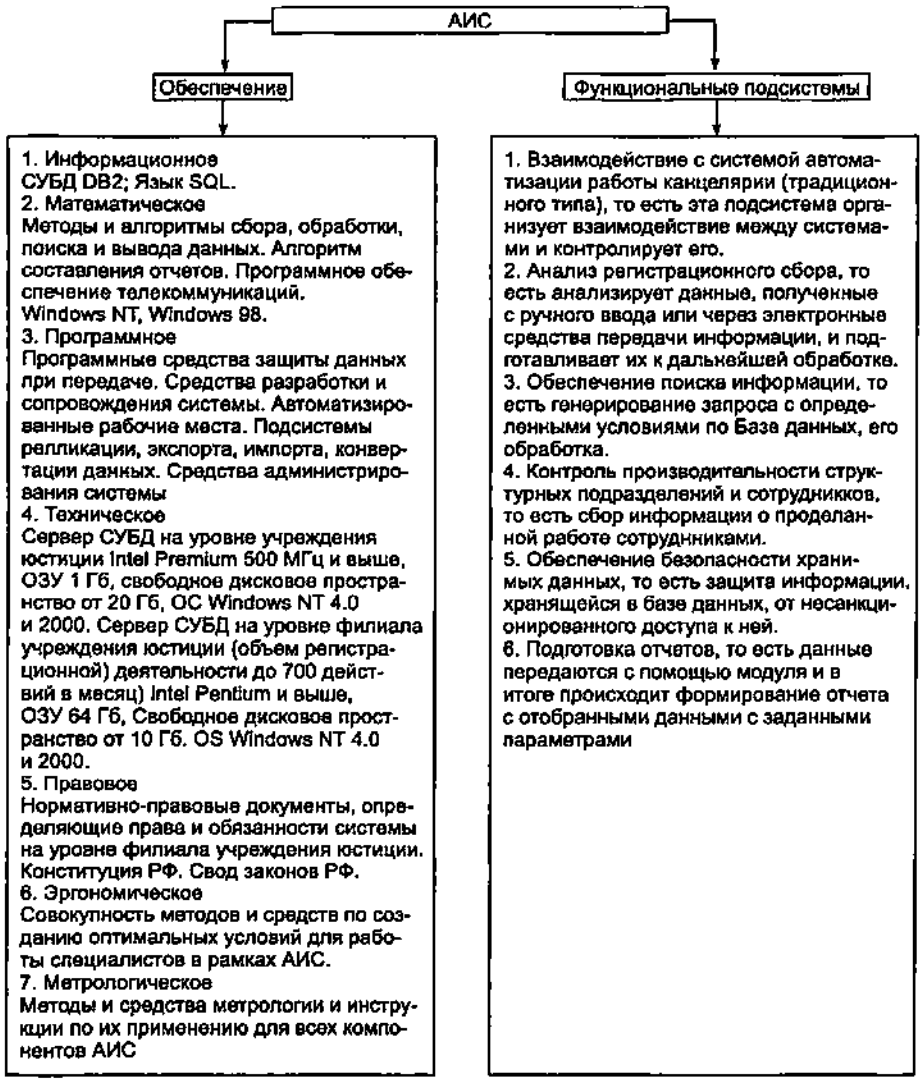


Рис. 9.23. Структура АИС «Реестр прав»

Технологический процесс обработки информации в системе включает:

1) подготовку исходной информации и ввод ее в вычислительный комплекс;

2) автоматическую обработку заказов, поступающих от потребителей;

3) выдачу документа «Отчет кассира»;

4) подготовку статистических отчетов и аналитических данных.

На рис. 9.24 представлена структура системы «Экспресс-3».

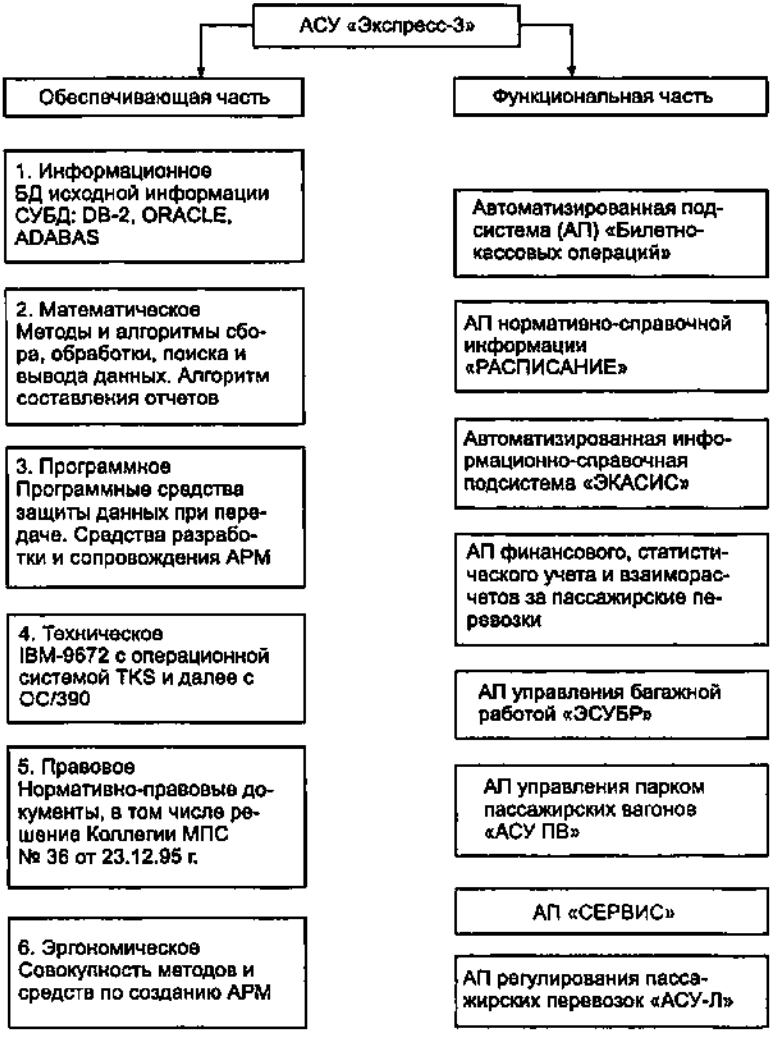


Рис. 9.24. Структура системы «Экспресс-3»

Информационное обеспечение. Исходная информация поступает в систему после печати проездного документа и содержит следующие показатели: дату и время отправления поезда, станции отправления и назначения, номер поезда, номер и тип вагона,

номер места, дату и время выдачи проездного документа, номер кассы и терминала, выдавших проездной документ.

Для обслуживания заказчика кассир обеспечивает связь через терминальное оборудование по мультиплексору передачи данных (МПД) с программой управления сообщениями. Запрос принимается программой формирования очереди и записи заявок. Заявку записывают на магнитный диск и формируют в пакет заявок. По программе связи осуществляется контроль сообщения и обработка пакета заявок. Изменения в результате выполнения запроса (заявки) заносят в основную базу данных и дубль базы данных, что позволяет поддерживать одинаковое состояние баз данных и избежать ошибок в работе. При сбое в работе оператор через консоль устанавливает оперативную связь с оператором системы для исправления неполадок.

Базы данных функционируют на базе следующих СУБД: DB-2, Oracle, ADABAS.

При формировании БД и работе с нею выполняются вышеуказанные технологические процессы.

Информационное обеспечение подсистемы «АСУ-Л» создается для дорожного и сетевого уровней. Дорожный уровень предполагает сбор, хранение, обработку информации в головном ВЦ «Экспресс-2» для формирования поездов. Сетевой уровень управления — информационные массивы формируются в региональных ВЦ «Экспресс-2», а затем по межмашинным запросам передаются в ВЦ «Экспресс-2» Московской железной дороги и на верхний уровень управления ЦЛ МПС.

Математическое и программное обеспечение. Математическое и программное обеспечение предусматривает использование типовых и разработку при необходимости новых методов и алгоритмов сбора, обработки, поиска и вывода данных и составления отчетов. Особое внимание уделяется разработке программных средств защиты данных при передаче их комплексу программ для разработки и сопровождения АРМ — автоматизированных рабочих мест.

Техническое обеспечение системы «Экспресс-3». Все микропроцессоры АСУ представляют единый вычислительный комплекс (ВК) системы, управляемый с одной консоли (пульта). Терминальное оборудование включает как старые терминалы АСУ «Экспресс-2», работающие по протоколу BSC-3, так и новые, работающие по двум протоколам: старому BSC-3 и новому X-25. Количество терминалов расширяется за счет установки их в при-

городных кассах, а также подключения ПЭВМ для выполнения ими разнообразных функций. Терминалы имеют специальное печатающее устройство. Автоматизируется весь финансовый и статистический учет, включая учет почасового пригородного пассажиропотока. Создается гибкое централизованное оперативное управление из АСУ «Экспресс-3» пригородными тарифами по дням недели и месяца.

Оборудование IBM-9672 работает на первом этапе под управлением операционной системы TKS, которая на последующих этапах будет заменяться на OC/390.

Правовое обеспечение. Нормативно-правовые документы, в том числе решение коллегии МПС № 36 от 23 декабря 1995 г., п. 3.5, о создании автоматизированных диспетчерских центров управления пассажирскими перевозками (АДЦУ) на дорогах и в МПС, обеспечивающих маркетинговый анализ рынка пассажирских перевозок и фирменное транспортное обслуживание пассажиров.

Функциональное обеспечение. Функциональное обеспечение системы включает комплекс подсистем:

1. Автоматизированная подсистема «Билетно-кассовых операций».
2. Автоматизированная информационно-справочная подсистема «ЭКАСИС».
3. Автоматизированная подсистема нормативно-справочной информации «РАСПИСАНИЕ».
4. Автоматизированная подсистема финансового, статистического учета и взаиморасчетов за пассажирские перевозки «ЭФИС».
5. Автоматизированная подсистема управления багажной работой «ЭСУБР».
6. Автоматизированная подсистема управления парком пассажирских вагонов «АСУ ПВ».
7. Автоматизированная подсистема «СЕРВИС».
8. Автоматизированная подсистема регулирования пассажирских перевозок «АСУ-Л».

Функциональная структура подсистемы «АСУ-Л» включает в себя комплекс экономико-математических, технологических, организационных методов, обеспечивающих решение задач оперативного регулирования и долгосрочного планирования таких перевозок.

Названия подсистем говорят об их назначении. Управление продажей билетов является важной частью организации пасса-

жирских перевозок. Широкое распространение нашли такие формы обслуживания, как оформление проездных документов от другой станции сети, на обратный выезд, с предварительным бронированием мест и заказы по телефону с доставкой на дом или по месту работы.

В оперативном режиме система позволяет получить информацию о динамике изменения населенности составов на маршруте следования в прямом и обратном направлениях, а также степень использования вместимости вагонов различных типов.

Функционирование подсистем, связанных с «населенностью», предусматривает решение задач четырех групп:

1. Накопление, обработка, формирование итоговых результатов об использовании вместимости вагонов и их хранение в ВК «Экспресс».

2. Передача информации о населенности вагонов в ПЭВМ.

3. Создание банка данных (летописи) о работе подвижного состава на сетевом и дорожном уровнях.

4. Реализация человеко-машинного диалога, который обеспечит:

- ввод с экрана указания режима работы;
- печать при необходимости выходных результатов на печатающем устройстве в виде таблиц установленной формы;
- обращение к летописи, корректировку ее информационных полей.

В функциональном плане новая система должна стать системой не только продажи билетов, но и управления всеми основными технологическими процессами пассажирского хозяйства, включая:

- билетно-кассовые операции, связанные с оформлением и учетом проездных документов во внутригосударственном, пригородном, межгосударственном и международном сообщениях;
- справочно-информационное обслуживание пассажиров; управление багажными и грузобагажными перевозками;
- эксплуатация и ремонт парка пассажирских вагонов;
- сервисное обслуживание пассажиров, включая поездки на других видах транспорта в смешанных поездках (автобусное, морское, речное, воздушное);
- экономика и финансовый учет по пассажирским перевозкам, включая взаиморасчеты между дорогами, контроль доходов и расходов;

- управление пассажирскими перевозками в целом по сети и отдельно по дорогам с помощью автоматизированных центров управления.

Контрольные вопросы

1. Каковы могут быть организационные формы функциональной части АИС?
2. Каковы функции подсистемы сбора информации?
3. Что такое интранет-технология?
4. Каковы функции подсистемы представления и обработки информации?
5. Что такое функциональные подсистемы АСУ? Приведите примеры.
6. В каком направлении идет развитие АСУ?
7. Какие вы знаете примеры функционирования АИС?

Глава 10
ТИПЫ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ И ПЕРСПЕКТИВЫ ИХ РАЗВИТИЯ

10.1. Типы АИС

Все системы можно классифицировать по различным классификационным признакам (под классификацией понимается разбиение множества на подмножества в соответствии с признаками сходства и различия), но среди множества классов есть специфический класс — АИС, — в котором представление, хранение и обработка информации осуществляется с помощью средств вычислительной и телекоммуникационной техники.

Любая классификация позволяет структурировать информационное пространство и локализовать определенную проблему с целью изучения взаимосвязей рассматриваемой проблемы с остальными элементами системы.

Классифицировать АИС по типам — это значит распределить их по определенным основаниям, характерным признакам. АИС классифицируют (типизируют) по разным основаниям:

- по назначению;
- по показателю условной информационной мощности;
- по степени автоматизации;
- по характеру представления и использования информации;
- по сфере применения;
- по территориальному (административному) делению;
- по масштабу;
- по способу организации;
- по видам выполняемых операций и др.

По назначению разрабатывают системы административные, общественные, политические, социальные, правовые, оборон-

ные, коммерческие, финансовые, образовательные, технологические, транспортные, связи и др.

По показателю условной информационной мощности (по количеству параметров) выделяют следующие системы:

- наименьшие (10—40);
- малые (41—160);
- средние (161—650);
- повышенные (651—2500);
- высокие (2501 и выше).

На рис. 10.1 представлена классификация АИС по следующим признакам:

- по степени автоматизации;
- по характеру используемой информации;
- по сфере применения.

В ручных ИС все операции по переработке информации выполняются человеком. Примером могут служить ручные картотеки, каталоги и т. д.

В автоматических ИС все операции по переработке информации выполняются на технических средствах без участия человека.

В автоматизированных ИС все операции по переработке информации выполняются в основном с помощью технических средств, но при участии человека.

По характеру представления, использования и логической организации информации выделяют АИС информационно-поисковые и справочные, которые включают фактографические, документальные, геоинформационные системы, а также информационно-решающие системы.

Информационно-поисковые системы (ИПС) выполняют ввод, систематизацию, хранение и выдачу информации по запросам. Сложное преобразование информации в таких системах не выполняется. ИПС могут быть документальными и фактографическими, т. е. работать или с документами, или с фактами из них.

В *документальных* ИС нерасчлененный документ является элементом информации, на входе — это входной документ. Информация при вводе, как правило, не структурируется или структурируется в ограниченном виде. Для вводимого документа могут устанавливаться некоторые формализованные позиции — вид, дата изготовления, назначение, тематика и т. д. Поиск документов в системе осуществляется по поисковому образу документа

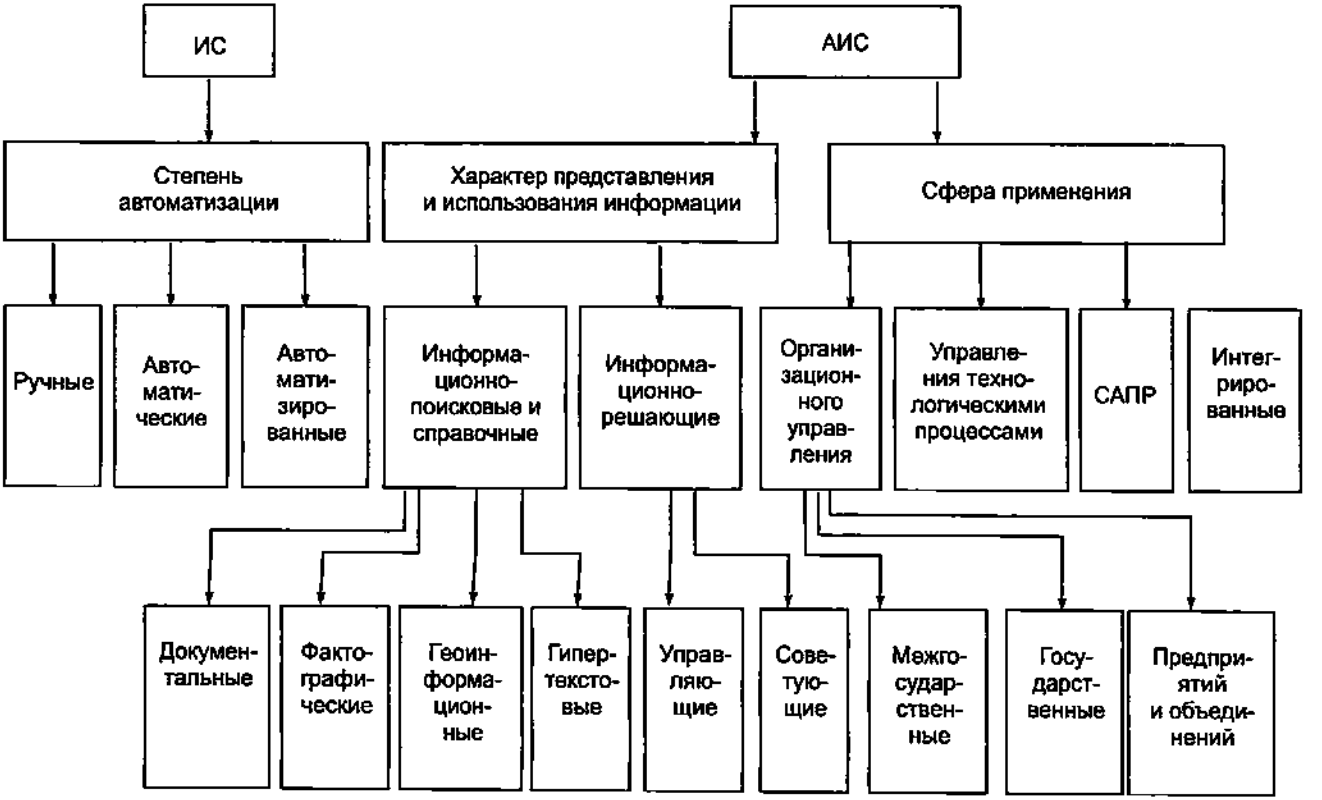


Рис. 10.1. Классификация АИС по степени автоматизации, характеру информации, сфере применения

(ПОД) — набору реквизитов (формализованных позиций), отражающих основные формализованные характеристики документа (вид, назначение, дата изготовления, тематика и т. п.).

Некоторые виды документальных АИС обеспечивают установление логической взаимосвязи вводимых документов — соподчиненность по смысловому содержанию, взаимные отсылки по каким-либо критериям и т. п. Определение и установление таких взаимосвязей представляет собой сложную многокритериальную и многоаспектную аналитическую задачу, которая не всегда может быть в полной мере формализована.

Примером документальных систем могут служить правовые системы «КонсультантПлюс», «Гарант», «Кодекс» и др., содержащие все нормативные документы по правовому законодательству страны, которые хранятся и регулярно обновляются.

Фактографические АИС накапливают и хранят данные в виде множества экземпляров одного или нескольких типов структурных элементов (информационных объектов). В фактографических ИС элементом информации является запись (агрегат) — данные (структурные элементы) об информационных объектах. Экземпляры структурных элементов или их совокупность дают сведения об отдельных фактах или их совокупности. По своей структуре каждый тип информационного объекта — набор реквизитов, характеризующий сведения об объектах АИС. Информация обязательно должна быть структурирована перед вводом в базу и занесена по определенным реквизитам.

Например, фактографическая АИС, построенная по принципу телефонной книги, каждому абоненту в базе данных ставит в соответствие запись, состоящую из набора таких реквизитов, как фамилия, имя, отчество, адрес проживания и номер телефона. Комплектование информационной базы данных в фактографических АИС включает, как правило, обязательный процесс структуризации входной информации из документального источника. Структуризация при этом осуществляется через выделение или вычисление экземпляров информационных объектов определенного типа, информация о которых имеется в документе, и заполнение их реквизитов.

Примером фактографических систем могут служить системы о личном составе любой организации, где о каждом сотруднике в базе накапливается информация по соответствующим реквизитам (фамилия, имя, отчество, год рождения и т. п.) и др.

В *геоинформационных системах* (ГИС) обрабатывается геодезическая, картографическая, статистическая, аэрокосмическая информация. Данные могут быть представлены в обычной (аналоговой) или цифровой форме. Данные организованы в виде отдельных информационных объектов с определенным набором реквизитов, привязанных к общей электронной топографической основе (электронной карте). Базы данных ГИС формируются на основе карт, представленных в цифровой форме. Цифровые карты служат основой для привязки (пространственного координирования) объектов и набора тематических слоев данных (лесные ресурсы, водные ресурсы, здания и сооружения и т. д.). Совокупность всех слоев образует общую информационную основу ГИС. ГИС классифицируют по следующим признакам: характеру модели, структуре модели баз данных (БД), особенностям модели интерфейса.

ГИС применяются для информационного обеспечения в тех предметных областях, структура информационных объектов и процессов в которых имеет пространственно-географический компонент, например маршруты транспорта, коммунальное хозяйство и т. д.

ГИС является пространственной информационной системой, общегеографической или отраслевой, и может быть:

- по тематике: социально-экономической, туризма, кадастровой, лесопользования, водных ресурсов, использования земель и др.;
- по территориальному признаку: общегосударственной или региональной;
- по целям: многоцелевой и специализированной.

Например, автоматизированная информационная система «Археограф» — информационная система обработки археологического материала на современном научном и методическом уровне. Структура АИС предусматривает регистрацию, учет и поиск информации, независимой от других АИС, а также ее непосредственное взаимодействие с ГИС с использованием топографических карт местности любого масштаба.

Имеется потребность в создании информационной системы не только для быстрого поиска необходимых данных, учета и хранения базовой информации о памятниках и находках (археологические карты), но также включающей в себя функции научно-исследовательской системы и функций контроля и охраны археологических памятников (паспорта на археологические па-

мятники). В силу своей специфики археологическая информация практически всегда должна иметь географическую привязку для анализа данных с учетом распространения материала во времени и пространстве.

Структура информационной системы предусматривает взаимодействие с геоинформационной системой для последующего анализа средствами ГИС. Система состоит из нескольких информационных модулей: названия, категории, административной и географической привязки, относительной и абсолютной хронологии памятника, кадастра, библиографии, архивных сведений и изобразительных материалов. Каждый модуль имеет свою структуру описания объекта. Особое внимание уделяется разработке взаимодействия программы с ГИС для последующего выполнения пространственного анализа данных, что позволяет оперировать большими объемами археологической информации с использованием цифровых карт местности за счет координатной привязки каждого археологического памятника.

Археологическая АИС расположена на центральном сервере ИИМК РАН и доступна в полном объеме в пределах локальной сети организации и в ограниченном объеме — через сеть Интернет. Структурированные данные об археологических памятниках и вся связанная с ними информация хранятся в файле интегрального формата СУБД НИКА, для работы с которым разработана специальная программная оболочка «Археограф». Взаимодействие системы с пользователем осуществляется с любого компьютера класса Pentium локальной сети ИИМК РАН через интерфейс программной оболочки, установленной на сервере под операционной системой Windows NT или через сеть Интернет через прикладную программу с веб-интерфейсом (просмотр и базовый поиск). Программная оболочка включает необходимые специализированные информационные модули, систему краткого и расширенного поиска, взаимодействия с ГИС через протокол DDE, экспорт данных.

Источником данных служит массив учтенных археологических памятников четырех регионов РФ: Карачаево-Черкесии, Красноярского края, Пермской области, Ленинградской области, общим количеством около 4200. В дальнейшем объем данных может превысить 10 тыс. за 2 года, что, по некоторым оценкам, может составить до 5—10 % выявленных памятников археологии на территории Российской Федерации.

В *гипертекстовых системах* поиск информации осуществляется по ссылкам (гиперссылкам) — выделенным цветом или подчеркнутым, связанным по смыслу фрагментам текста того же или другого документа.

Гипертекст — набор страниц, организованных в некоторую последовательность при помощи ссылок.

Активное развитие технологий текстового поиска стимулировало создание поисковых систем более общего класса, которые имеют дело не только с текстовыми документами, но и с информацией, представленной в различных иных средах. В таких системах (их называют мультимедийными) содержание объектов поиска представляет собой сочетание текстовых элементов, статических изображений, музыкальных произведений, мультфильмов, видеоклипов и т. п.

За свою почти полувековую историю развития технологии текстового поиска продвинулись от простейших дескрипторных информационно-поисковых систем к изощренным системам полнотекстового поиска, от поисковых систем к системам с более богатой функциональностью. Технические возможности современных вычислительных систем позволяют хранить огромные объемы информации в системах текстового поиска, осуществлять в них тонкую обработку документов, выполнять алгоритмически сложные процедуры обработки хранимых коллекций документов — их классификацию, кластеризацию, глубинный анализ текстов, перевод документов с одного языка на другой и т. д.

Новое зарождающееся направление текстового поиска связано с потоками документов. Поскольку, в отличие от предположений, положенных в основу уже разработанных технологий, здесь отсутствуют какие-либо сведения, характеризующие коллекцию документов в целом, для поиска в потоке документов необходимы новые подходы и новые методы.

Системы текстового поиска оказали значительное влияние на формирование специфического класса АИС, называемых системами управления документами, которые широко используются в настоящее время во многих крупных коммерческих компаниях и в других организациях. В таких системах важная роль отводится не только методам обработки естественного языка, созданным для работы с текстовыми документами, но и организации групповой разработки документов, их хранения, распространения, а также технологиям текстового поиска.

Текстовый поиск, как уже отмечалось, имеет почти полувековую историю. Рассмотрим кратко некоторые наиболее значимые вехи его развития.

Ранние системы. Ранние текстовые АИС были ориентированы на функцию поиска, поэтому назывались информационно-поисковыми системами (ИПС). Разработки простейших систем такого рода — дескрипторных ИПС — велись еще в 1950-х годах. Одной из наиболее распространенных областей их применения был библиографический поиск. Многие дескрипторные библиографические ИПС используются до настоящего времени.

Большое значение для дальнейшего развития технологий текстового поиска имели исследования и разработки экспериментальных прототипов, выполненные в 1960-х годах. На этом этапе начали формироваться технологии полнотекстового поиска, т. е. поиска по полному содержанию текстовых документов, представленных на естественных языках.

Прежде всего получил распространение контекстный поиск. Так называется поиск документов, имеющих вхождение в них заданного контекста. Позднее были созданы методы контекстного поиска, учитывающие грамматические формы элементов контекста, фонетическую близость слов и т. д.

Наряду с техникой контекстного поиска в этот период были предложены новые подходы к построению поисковых систем, обеспечивающие существенное повышение качества поиска, особенно в сети Интернет.

Документальным массивом ИПС Интернет является все множество документов. Это довольно разнородная информация, которая представлена в виде различных, несогласованных друг с другом форматов данных: тексты, графическая и аудиоинформация и др. Это могут быть веб-страницы, Gopher-файлы, документы Wais, записи архивов FTP, новости Usenet, статьи почтовых списков рассылки.

В традиционных системах при поиске используется поисковый образ документа (ПОД), используемый вместо реального документа. ПОД является результатом применения некоторой модели информационного массива документов к реальному массиву. Первая задача, которую должна решить ИПС, — это приписывание списка ключевых слов документу или информационному ресурсу. Эта процедура и называется индексированием. Проблема, связанная с индексированием, заключается в том, что приписывание поискового образа документу или информацион-

ному ресурсу опирается на словарь, из которого эти термины выбираются. В традиционных системах существовало разбиение на системы с контролируемым словарем и системы со свободным словарем. Контролируемый словарь предполагал ведение некоторой лексической базы данных. Свободный словарь пополняется автоматически по мере появления новых документов. В момент обновления перегружаются сами документы и обновляется словарь, а после его обновления производится переиндексация документов. Для решения этой задачи используются программы сканирования сети или роботы-индексировщики. Робот просматривает Сеть, находит новые ресурсы, приписывает им термины и помещает в базу данных индекса. Роботы обычно используют для индексирования следующие источники для пополнения своих виртуальных словарей: гипертекстовые ссылки, заголовки, заглавия, аннотации, списки ключевых слов, полные тексты документов, а также сообщения администраторов о своих веб-страницах.

После того как ресурсы индексированы и система составила массив ПОД, начинается построение поискового аппарата. Для ускорения поиска строится индекс, которым в большинстве систем является набор связанных между собой файлов, ориентированных на быстрый поиск данных по запросу. Структура и состав индексов различных систем зависят от многих факторов: размера массива поисковых образов, информационно-поискового языка, размещения различных компонентов системы и т. п. Например, индекс системы Yahoo! состоит из таблицы идентификаторов страниц (Page-ID), таблицы ключевых слов (Keyword-ID), таблицы модификации страниц, таблицы заголовков, таблицы гипертекстовых связей, инвертированного (IL) и прямого списка (FL) — массива поискового образа страниц.

Результат поиска в данном файле — это объединение и/или пересечение списков идентификаторов страниц.

Информационно-поисковый язык системы. Индекс — это только часть поискового аппарата, скрытая от пользователя.

Второй частью этого аппарата является информационно-поисковый язык (ИПЯ), позволяющий сформулировать запрос к системе в простой и наглядной форме. Обычно фраза разбивается на слова, из которых удаляются запрещенные и общие слова, иногда производится нормализация лексики, а затем все слова связываются либо логическим И, либо ИЛИ. Возможны и другие варианты. Наиболее развитым языком запросов из современ-

ных ИПС обладает AltaVista. Эта система позволяет использовать еще и связку NEAR (Близкие), позволяющую организовать контекстный поиск. Все документы в системе разбиты на поля, поэтому в запросе можно указать, в какой части документа пользователь надеется увидеть ключевое слово: в ссылке, заглавии, аннотации и т. п.

При составлении запроса к системе используют либо меню-ориентированный подход, либо командную строку. Первый подход позволяет ввести список терминов, обычно разделяемых пробелом, и выбрать тип логической связи между ними. Логическая связь распространяется на все термины.

Результат поиска в базе данных ИПС — это список указателей на удовлетворяющие запросу документы. Различные системы представляют этот список по-разному. В Yahoo!, например, это количество терминов запроса, содержащихся в ПОД, в соответствии с которым ранжируется результат поиска. Важна и процедура коррекции запросов по релевантности. Релевантность — это мера соответствия найденного системой документа потребности пользователя. Различают формальную релевантность и реальную. Первую вычисляет система, на основании чего и ранжируется выборка найденных документов. Вторая — это оценка самим пользователем найденных документов.

Сегодня практически нельзя найти что-либо полезное в Интернете без услуг информационно-поисковых систем (ИПС), имеющих уже двадцатилетнюю историю, например таких, как AltaVista, Yahoo!, Open Text и др. Результат поиска в терабайтных массивах информации осуществляется достаточно быстро, но только при правильном планировании стратегии поиска и знакомстве с основными положениями теории ИПС. На сегодняшний день нет другого способа быстрого поиска данных, кроме поиска по ключевым словам.

AltaVista появилась в конце 1995 г. и работала на 64-битовом вычислительном алфавитном сервере DEC (разработана корпорацией AltaVista Technology, но продана Digital Equipment Corporation; в 1998 г. ее купила корпорация Compaq, а в 1999 г. — CMGI). Эта самая быстрая поисковая система обрабатывала миллионы поисковых запросов в день на естественном языке (рис. 10.2). Адрес поисковой системы — altavista.digital.com.

На рисунке использованы следующие обозначения.

Client (клиент) — программа просмотра конкретного информационного ресурса. Наиболее популярны сегодня мультипрото-

кольные программы типа Netscape Navigator. Такая программа обеспечивает просмотр документов Сети, FTP-архивов, почтовых списков рассылки и групп новостей. Все эти информационные ресурсы также являются объектом поиска ИПС.

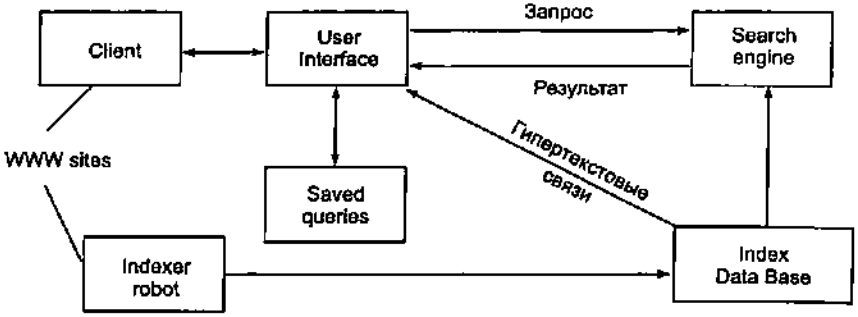


Рис. 10.2. ИПС AltaVista

User Interface (пользовательский интерфейс) — не просто программа просмотра, а также способ общения пользователя с поисковым аппаратом: системой формирования запросов и просмотров результатов поиска.

Search Engine (поисковая машина) — служит для трансляции запроса на информационно-поисковом языке (ИПЯ) в формальный запрос системы, поиска ссылок на информационные ресурсы Сети и выдачи результатов поиска.

Index Database (индекс базы данных) — индекс, который является основным массивом данных ИПС и служит для поиска адреса информационного ресурса. Архитектура индекса для максимально быстрого поиска. Происходит при этом оценка ценности каждого из найденных информационных ресурсов Сети.

Queries (запросы пользователя) — сохраняются в пользовательской личной базе данных. На запросы уходит достаточно много времени, но система их запоминает и дает хорошие ответы.

Indexer Robot (робот-индексировщик) — служит для сканирования Сети и поддержания базы данных индекса в актуальном состоянии. Эта программа является основным источником информации о состоянии информационных ресурсов Сети.

WWW Sites — весь Интернет, информационные ресурсы, просмотр которых обеспечивается программами просмотра.

Большой популярностью в Интернете пользуется международная система *Google*.

Рассмотрим схему функционирования Google. Всю основную работу по просеиванию содержимого Сети выполняют интернет-роботы. Каждый из них берет один адрес (URL — локатор ресурсов, каждый соответствует определенному идентификатору документа) из базы данных URL-сервера, скачивает и передает содержимое страницы на сервер хранения документов (рис. 10.3). Все содержимое сервера хранится в заархивированном виде для увеличения его вместимости.

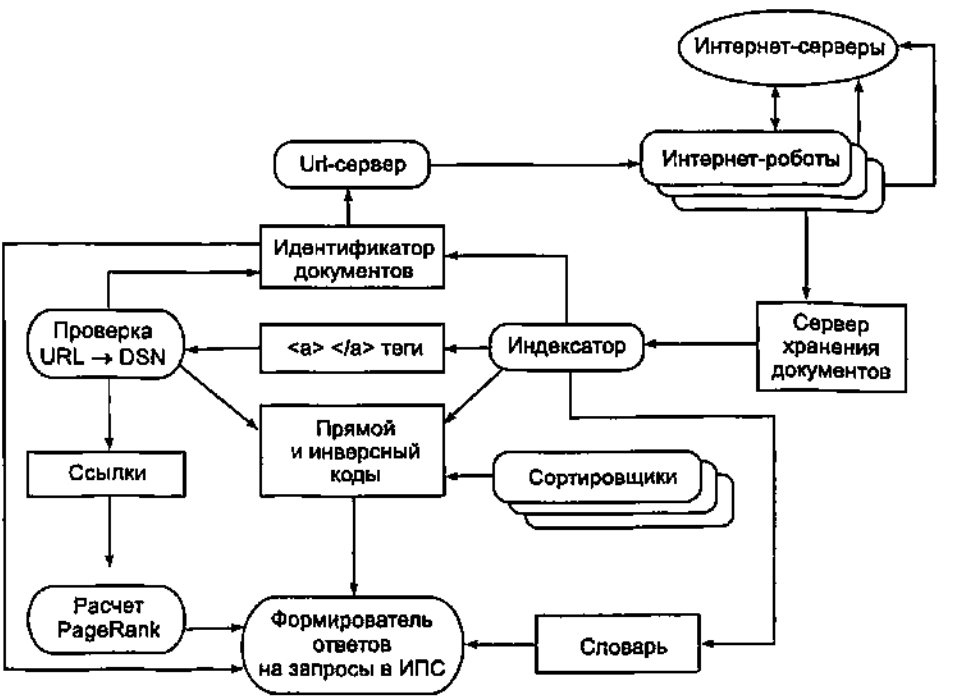


Рис. 10.3. Структура информационно-поисковой системы Google

Программа-индексатор разлагает текст документа на составляющие его слова (хит в терминологии Google), запоминая при этом местонахождение, шрифтовой вес, а также написано ли слово заглавными или строчными буквами и принадлежит ли оно к категории «особенных» (названия документов, метатеги, URL и тексты ссылок). Вся эта информация складывается в набор контейнеров, именуемых прямым индексом.

Идентификаторы слов берутся из словаря, который постоянно пополняется. Одновременно с этим индексатор просматрива-

ет содержимое тегов и проверяет корректность всех ссылок в службе доменных имен DNS (Domain Name Service).

Каждая из программ, обозначенных на рис. 10.3 прямоугольником со скругленными углами, работает независимо от других, причем аппаратные конфигурации серверов и рабочих станций, на которых функционирует механизм Google, выбираются так, чтобы не создавать «пробок» при обработке информации, собранной интернет-роботами.

Структура прямого индекса не очень удобна при поиске документов на основании встречающихся в них слов (пользователь задает слово или словосочетание, а система должна найти подходящий документ). Чтобы решить эту проблему, был введен так называемый инверсный, или обратный, индекс. В нем любому слову из словаря соответствует набор doc_id-документов, в которых это слово встречается. Работой по постоянному формированию инверсного индекса занимаются сортировщики. Так как, во-первых, всегда появляются новые документы и, во-вторых, обновляются старые, индекс приходится постоянно перестраивать.

Пусть от пользователя поступил запрос найти документы со словом «информация». Программа, формирующая ответы, посмотрит в словарь, найдет там идентификатор для слова «информация», сформирует запрос в базу данных с использованием инверсного индекса и получит набор документов, в которых это слово встречается. Далее на основании значения PageRank количества хитов, их качества и, может быть, других ограничений и приоритетов разработчиков будут распределены порядковые номера страниц в выходном списке. В итоге пользователь Интернета получит самую оптимальную, по мнению ИПС, информацию о том, где и что писали о правилах и способах получения информации.

Информационно-решающие системы (*ИРС*) по определенным алгоритмам выполняют переработку информации, воздействуя полученными результатами на процесс принятия решений. ИРС можно подразделить на управляющие и советующие.

Управляющие системы, как правило, обрабатывают большие объемы информации и производят в основном информацию расчетного характера, помогающую человеку принимать решения. К ним относятся так называемые бухгалтерские конструкторы, т. е. бухгалтерские системы с расширенными инструментальными возможностями. Пользователь, овладев специальным

языком, может научить программу выполнять необходимые ему расчеты, создавать отчеты. Имеющиеся универсальные программы можно с помощью настроек превращать в программы, которые подходят для расчетных работ в любой организации.

Примером управляющих систем могут служить бухгалтерские конструкторы «1С:Бухгалтерия», «Инотек», Quiken и др. Так, с помощью программы «1С:Бухгалтерия» бухгалтер может настраивать план счетов, виды первичных и отчетных документов, схемы проводок, имеет возможность ведения сложного аналитического учета.

Управляющие финансово-аналитические системы дают возможность проводить анализ хозяйственной деятельности предприятия и/или работать с инвестиционными проектами. Примерами таких систем могут быть система бизнес-планирования Project Expert, Budget management и др.

Применение начинают находить «эккаунт кутюр» — системы, индивидуально дорабатываемые на базе типового бухгалтерского ядра для конкретного заказчика и внедряемые разработчиком на конкретном предприятии.

Советующие системы обрабатывают уже не только данные, но и знания. Они являются системами с достаточно высокой степенью интеллекта. Выдаваемая ими информация рекомендуется для принятия к сведению, а не для выполнения конкретных операций. Такие системы обладают богатейшими возможностями для автоматизации интеллектуального труда (например, для составления справочников, словарей и т. п.). В эти системы можно добавлять любую новую информацию.

Высшим этапом развития советующих систем явились *экспертные системы* (ЭС). Их идеологию выражают формулой ЭС = = знание + вывод. Проблемы ставятся перед системой в виде совокупности фактов, а она с помощью БЗ выводит заключение из этих фактов. Экспертная система имеет взаимодействующие блоки (основные: база знаний и механизм вывода). Представление знаний базируется на процессе распознавания, т. е. на обучении и собственно распознавании. В процессе обучения обрабатываются данные многочисленных наблюдений над объектами. Выявляются закономерности, присущие данному классу объектов, и на их основе строят формализованную модель базы знаний. На основе полученных результатов выводят определенные решающие правила. В процессе собственно распознавания правила применяются для интересующих объектов данного класса, кото-

рые непосредственно не измеряются. В ЭС известен алгоритм обработки знаний, а не алгоритм решения задач. Алгоритм обработки знаний строится по ходу решения задачи на основании эвристических (найденных методами проведения аналогий и абстрагирования) правил, которые хранятся в БЗ. Цикл функционирования системы состоит в следующем: выбор (запрос) данных или результатов анализа — наблюдения — интерпретация результатов — усвоение новой информации — выдвижение гипотез — выбор следующей совокупности данных. Цикл повторяется до того времени, пока не будет достаточно информации для заключения.

Экспертная система — набор программ, разработанных на языках высокого уровня. При их написании используются как традиционные языки Паскаль, Си и др., так и языки искусственного интеллекта Лисп, Пролог и др., а также специальные средства поддержки разработки. Они имеют некоторые отличия в структуре и характеристиках по сравнению с обычными программами и осуществляют функции эксперта при решении задач в области его компетенции. ЭС выдают советы, проводят анализ, классификацию, дают консультации.

Эти системы путем привлечения эвристик часто находят решение задач, которые неструктурированны и плохо определены.

В настоящее время разрабатывают «пустые оболочки» ЭС с различными моделями представления знаний и различными механизмами логического вывода, которые можно использовать для внесения новых знаний в разных предметных областях. Задача пользователя состоит не в непосредственном программировании, а в формализации и вводе знаний с использованием предоставленных оболочкой возможностей. Недостатком этих оболочек можно считать невозможность охвата одной системой всех существующих предметных областей. Примеры: InterExpert, PC+, VP-Expert. Имеются генераторы экспертных систем — мощные программные продукты, предназначенные для получения оболочек, ориентированных на представление знаний в зависимости от рассматриваемой предметной области. Примеры: KEE, ART и др.

ЭС нашли широкое применение в медицинской диагностике, прогнозировании, планировании, интерпретации результатов наблюдений. Системы используются при контроле и управлении в различных сферах деятельности, например при проектировании космических станций, управлении атомными станциями,

воздушным движением и т. д. Системы применяют при диагностике неисправностей в механических и электрических устройствах, например при ремонте автомобилей, локомотивов, компьютеров и т. д.

Используют при обучении — компьютерные игры, обучающие программы. Применяют системы в прогнозировании погоды, урожайности, потоков пассажиров и т. д.

Можно назвать следующие системы: в медицине — MYCIN, Dendral, в геологии — PROSPECTOR, в компьютерной технике — XCON, XSEL, в судовождении — HASP/SIAP, в обучении — EURISCO. Наиболее распространены в настоящее время за рубежом экспертные системы и их оболочки Insight, Logian, Nexpert, Rule Master, KDS, Picon, Knowledge Craft, KESII, SI, TIMM и др.

К недостаткам ЭС можно отнести недостаточную простоту в использовании, невозможность формализовать абсолютно все знания, а также то, что они не обладают здравым смыслом и правила БЗ меняются чаще (с накоплением опыта), чем алгоритмы, поэтому их следует чаще модифицировать, и др.

Следует отметить следующие преимущества ЭС перед экспертом: у них нет предубеждений, они не делают поспешных выводов, делают выводы на основе многочисленных наблюдений, БД хранят огромное количество знаний многих экспертов. ЭС являются эффективным инструментом для принятия правильных решений человеком.

Современные ЭС тесно переплетаются с существующими бизнес-системами. Объединение их компонентов, особенно БЗ и ПО, позволяет существенно сократить расходы на подготовку квалифицированного персонала, уменьшить другие расходы. Часто ЭС объединяют с САПР. Приходят к выводу, что системы с БЗ необходимо встраивать в самые важные бизнес-процессы.

Информационные системы организационного управления осуществляют автоматизацию функций управленческого персонала. Такие системы могут предназначаться как для отраслевого, территориального и общегосударственного управления, так и для управления персоналом различных организаций (промышленных предприятий, торговых фирм, банков и т. д.).

Например, на уровне хозяйства всей страны действуют *общегосударственные* (межотраслевые) автоматизированные системы: плановых расчетов — АСПР; государственной статистики — АСГС; управления трудовыми ресурсами и т. п.

Например, АИС «Налог» представляет собой систему организационного управления органами Госналогслужбы. АИС при минимальных затратах ручного труда должна обеспечить сбор, обработку и анализ информации о состоянии объекта управления, выработку управляющих воздействий, обмен информацией внутри системы и между другими системами, которые с ней взаимодействуют (например, «КонсультантПлюс», «Гарант» и др.).

С другой стороны, АИС этого типа можно классифицировать *по видам выполняемых операций*:

- *информационно-измерительные системы* (*ИИС*) обеспечивают автоматизацию сбора и регистрации информации о состоянии элементов наблюдаемых процессов;
- *информационно-поисковые системы* (*ИПС*) обеспечивают выполнение поисковых операций. Выделяют ИПС документального типа, фактографического типа и комбинированные, сочетающие возможности документального и фактографического поиска;
- *информационно-справочные системы* (*ИСС*) обеспечивают поиск и различные виды обработки информации с целью информирования пользователя о состоянии системы для формирования решений по управлению объектом;
- *информационно-советующие системы* (*ИСоС*) обеспечивают формирование множества альтернатив принятия решений по управлению объектом. Лицо, принимающее решение (ЛПР), выбирает конкретный вариант управляющего воздействия из предложенных альтернатив;
- *информационно-управляющие системы* (*ИУС*) обеспечивают формирование оптимального варианта управляющих воздействий, который передается ЛПР либо непосредственно на управляемый элемент системы.

Приведенная классификация учитывает только специфику выполняемых операций, а на реальном объекте могут использоваться различные сочетания ИС для перекрытия потребностей системы. Например, в крупной библиотеке базовой системой (реализующей основную функцию объекта) будет ИПС, дополнительно необходима ИСС, обеспечивающая сбор статистики об использовании фондов, о затратах на содержание библиотеки и т. д. Если необходимо управлять условиями хранения библиотечного фонда, потребуются ИИС и ИУС, обеспечивающие сбор сведений об условиях хранения и поддержания этих параметров в заданном диапазоне.

По территориальному (административному) признаку различают следующие системы:

- общероссийские;
- областные (краевые);
- общереспубликанские;
- городские.

В них решаются задачи баланса производительных отраслей хозяйства и всех необходимых видов ресурсов на территории, их рационального использования, создания различных предприятий и т. д.

Например, автоматизированная информационная система управления органов ЗАГС (записи актов гражданского состояния) Ставропольского края (разработчик: компания КРОК) предусматривает внедрение информационных технологий во все краевые органы исполнительной власти. Система разработана в период 2002—2005 гг. и имеет центральный офис (в городе Ставрополе) и 33 удаленных офиса (отделы ЗАГС). Сейчас система развернута на 35 коммуникационных серверах и серверах баз данных и имеет не менее 180 рабочих мест. Создан электронный архив фонда записей гражданского состояния, обеспечивается его сохранность. Система осуществляет поиск, дополнения и выдачу информации о людях и записей актов. Реализован простой и удобный интерфейс, являющийся единым для всех подсистем.

АИС ЗАГС построена на основе интернет-технологий, имеет трехуровневую архитектуру с выделенным сервером приложений MS Internet Information Server. Специалисты-разработчики считают, что такое решение является шагом вперед по сравнению с архитектурой клиент/сервер. Основная идея технологии — разделение ключевых функций по обработке информации между программой приложения клиентов и программой СУБД. На сервер возлагается большая часть обязанностей по оптимизации обслуживания, поддержке целостности, безопасности данных, контролю.

В управлении ЗАГС сформирована БД на основе получаемых из отделов ЗАГС сведений. На сервере сосредоточена централизованная обработка данных, что обеспечивает высокую производительность, надежность хранения информации, эффективность обработки данных и их защиту. Обмен данными между подразделениями всех уровней осуществляется с помощью документов (электронных сообщений) в формате XML по протоколу SMTP. В вышестоящих организациях происходит автоматическое объ-

единение информации путем интеграции АИС с почтовым сервером MS Exchange Server. В отделах ЗАГС есть все необходимое для автономной работы. Обмен данными между отделами ЗАГС и управлением может быть налажен по коммутируемым линиям связи с помощью модемов и почтовой программы MS Outlook Express, без использования выделенных линий связи. Многоуровневая защита АИС базируется на встроенных средствах программного обеспечения. Работа конечных пользователей происходит в среде веб-обозревателя MS Internet Explorer версии 5.5 и выше.

Опыт реализации аналогичных проектов имеется в Москве, Астраханской, Ивановской, Новгородской и других областях.

Можно выделить класс *экономических* АИС. К экономическим АИС относятся:

- отраслевые АИС;
- промышленных предприятий;
- банковские АИС;
- АИС рынка ценных бумаг и др.

Специфика данного класса заключается в использовании экономических (хозяйственных, производственных) параметров при отражении состояния элементов системы. Любой экономический объект состоит из двух основных частей: управляемого процесса и управляющей системы. Различают одноуровневые и многоуровневые многоцелевые системы управления. Все процессы, протекающие на объекте, могут быть разделены по видам обрабатываемых предметов (предметная специализация), по видам выполняемых операций обработки (технологическая специализация). Управляющие звенья по видам деятельности производят декомпозицию главной цели на подцели в зависимости от специфики вида деятельности, т. е. формируется вектор целей по видам деятельности, обеспечивающий достижение главной цели системы. Сформированные подцели доводятся до всех участников процесса. Управляющее звено логически законченной группы процессов или операций разрабатывает технологию достижения сформированных целей, которая по обратным связям согласуется со звеном управления по видам деятельности. Технология достижения целей передается управляюшему звену.

Информационные *отраслевые* АСУ создаются для управления соответствующей отраслью хозяйства: ОАСУ строительства — ОАСУС, АСУ транспорта — АСУТ и т. д.

Характерной особенностью организации систем является то, что объектом управления в них являются люди, коллективы людей. В таких системах управление производственными процессами осуществляется как процесс косвенного или непосредственного воздействия на людей, на предметы труда управления производством. Управление рассматривается как процессы, обеспечивающие согласованность и целенаправленность производительной деятельности коллективов людей, которые, в свою очередь, управляют средствами труда. Предмет труда и продукт труда в таких системах — информация. Важнейшая задача организации таких систем (АСУ) — совершенствование управления экономикой на всех уровнях. В АСУ решаются два класса задач: стратегические — выбор связей между системами; анализ, оценка поведения системы; тактические — реализация планов и стратегий.

Информационные АСУ должны обеспечивать:

- автоматизированный сбор и обработку информации с широким использованием метода оптимизации;
- хранение и комплексное использование нормативно-справочной, оперативной и другой необходимой информации для принятия решений;
- рациональный документооборот;
- внедрение прогрессивных методов планирования учета и анализа.

Для этого в состав системы должны входить:

1. ЭВМ соответствующего уровня и другое необходимое оборудование.

2. Программные средства, обеспечивающие выполнение необходимых расчетов.

3. Рациональные методы планирования и управления.

В состав отрасли информационной ОАСУ должны входить следующие подсистемы:

- перспективного развития отрасли;
- планирования;
- оперативного управления;
- управления сбытом продукции;
- управления финансовой деятельностью;
- управления учетом и анализом труда и заработной платы;
- управления сбытом продукции;
- учета анализа кадров;
- управления НИР;
- управления капитальным строительством;

- управления бухгалтерским учетом;
- научно-технической информации.

Широкое распространение нашли отраслевые системы бухгалтерского учета, построенные на основе бухгалтерского комплекса АРМ, к которому подсоединены специализированные отраслевые АРМ.

В качестве примера отраслевых систем можно привести системы «Торговля», «Бюджетные организации», «Промышленность», «Строительство», «Аудит», «Банковские структуры», «Страхование» и др.

Организационные системы управления предприятиями (*АСУП*) предназначены для оперативного контроля, учета и анализа, планирования, управления материально-техническим снабжением и решения многих других организационных и экономических задач в конкретной организации.

Цель информационного управления в системе может быть достигнута путем выполнения определенных функций управления: планирования; организации; регулирования; контроля и учета.

Планирование — это определение цели управления и путей ее достижения, определение плана действий, прогнозирования.

Организация — выбор и форматирование структуры системы управления, определение соотношения между ее элементами и их взаимодействия.

Регулирование — это поддержание требуемого соотношения между различными элементами системы, ликвидация возможных отклонений от плановых заданий.

Контроль — наблюдения и проверка соответствия действительного и планового хода процесса производства.

Учет — подтверждение итогов выполнения плана или отдельных этапов его осуществления, оценка результатов управления.

Примером может служить отраслевая интегрированная автоматизированная информационная система комбината хлебопродуктов — АИС КХП (разработана МГУ пищевых производств и фирмой «ИНФО»).

Система создана для использования непосредственно на крупных и средних производственных предприятиях хлебоприемной и зерноперерабатывающей промышленности и может быть настроена для эксплуатации на любом предприятии близких перерабатывающих отраслей (масложировой, солодовой, хлебопекарной и др.). На программах АИС КХП работают свы-

ше 50 предприятий отрасли хлебопродуктов в России и ближнем зарубежье. АИС КХП постоянно совершенствуется и развивается.

Информационные автоматизированные системы управления технологическими процессами (*АСУТП*) осуществляют автоматизацию функций управления производственным персоналом и различными технологическими процессами в химической, металлургической, машиностроительной промышленности и др.

Например, в качестве средств автоматизации процессов изготовления деталей используют оборудование с ЧПУ, АЛ (автоматизированные линии), ГПС (гибкие производственные линии), ПР (промышленные роботы), РТК (роботизированные технологические комплексы).

Проектирование технологических процессов обработки заготовок деталей на АЛ начинают с анализа информации о продукции. Технологический процесс разрабатывают поэтапно. Необходимым элементом перехода от локальной автоматизации к ГПС являются РТК, они могут или функционировать как самостоятельный вид оборудования, или быть объединены в роботизированные технологические участки (РТУ).

Структура РТК и степень участия человека в производственном процессе зависят от уровня автоматизации, избираемого для данных конкретных условий, и характера связей с внешними и смежными производственными подразделениями.

Связи как внутри РТК, так и между ними определяют характер выполняемого процесса. РТК можно условно подразделить на три основные подсистемы: обработки; обслуживания; контроля и управления.

Подсистема обработки формируется на основе технологического оборудования, модернизированного для взаимодействия с ПР; *подсистема обслуживания* формируется на основе устройств в РТК, межоперационных транспортирующих устройств и накопителей, устройств приема деталей, а также ПР.

Система контроля и управления состоит из комплекса программ средств контроля, измерений регулирования вычислений, логического управления, регистрации и аварийной защиты. Эта система оснащена датчиками контроля ориентации и точности расположения изготавливаемых изделий, выполняемого технологического процесса, правильности срабатывания оборудования, приспособлений, режущего инструмента, вспомогательными устройствами и средствами аварийной защиты.

Сложные РТК могут иметь СИО (систему инструментального обеспечения), технического обслуживания (ТО) и организационно-технического управления (ОТУ). Простые РТК используются как устройства управления.

Типичным примером АСУТП в связи может служить автоматическая коммутационная станция. В этой системе управление осуществляется с помощью технических устройств типа процессоров или других более простых приборов. Человек-оператор не входит в контур управления, замыкающий связи объекта и органа управления, а лишь следит за ходом технологического процесса и по мере необходимости (например, в случае сбоя) вмешивается.

Иначе обстоит дело с автоматизированной системой управления производственным процессом (АСУПП). В системе и объект, и орган управления представляют собой единую человеко-машинную систему. Человек обязательно входит в контур управления. АСУПП — это человеко-машинная система, предназначенная для сбора и обработки информации, необходимой для управления производственным процессом, т. е. управления коллективами людей в процессе производства. Успех функционирования таких систем во многом зависит от свойств и особенностей жизнедеятельности людей. Без человека АСУПП самостоятельно не может работать, так как человек формирует задачи, разрабатывает все виды обеспечивающих подсистем, выбирает из выданных ЭВМ вариантов решений наиболее рациональный. Человек, что очень важно, юридически отвечает за Нрезультат реализации принятых им решений. Роль человека огромна и незаменима. Человек организует программу подготовительных мероприятий перед созданием системы, следовательно, требуется помимо всего прочего специальное организационное и правовое обеспечение.

САПР — информационные системы автоматизированного проектирования. Их применяют для автоматизации функций специалистов различного профиля (проектировщиков, конструкторов, архитекторов и др.), т. е. для производства расчетов, чертежей, схем, планов, при моделировании объектов и т. п., при создании новой техники и технологии. САПР опирается на прочную научно-техническую базу: современные средства вычислительной техники, новые способы представления и обработки информации, создание новых численных методов решения инженерных задач и оптимизации. Системы автоматизированного проектирования дают возможность на основе новейших

достижений фундаментальных наук отрабатывать и совершенствовать методологию проектирования, стимулировать развитие математической теории проектирования сложных систем и объектов.

Выделяют САПР-конструктора и САПР-технолога. САПР базируются на АРМ (автоматизированные рабочие места) специалистов в самых различных сферах деятельности. Например, с помощью САПР проектируют различные компоновки ГАЛ (гибких автоматизированных линий) и др.

Говоря об АРМ, следует отметить, что они могут являться и базой для бухгалтерских комплексов, где для каждого раздела учета создаются отдельные программы с учетом последующего агрегирования данных. Пример бухгалтерского комплекса — АРМ «Монолит» и др. Таким образом, АРМ — это профессионально-ориентированные малые вычислительные системы, расположенные на рабочих местах специалистов и предназначенные для автоматизации их работ.

В настоящее время широкое применение нашли коммерческие информационные системы (КИС), имеющие следующую классификацию: класс А, класс В, класс С.

Класс А — системы (подсистемы) контроля и управления технологическими объектами и/или процессами. Объектами контроля в таких системах являются технологическое оборудование, датчики, исполнительные устройства и механизмы. Системы имеют контур обратной связи, малый временной интервал обработки данных и др.

В качестве примера систем класса А можно привести систему диспетчерского контроля и накопления данных — SCADA (Supervisory Control And Data Acquisition), распределенные системы управления — DCS (Distributed Control Systems), системы последовательного управления — Batch Control, автоматизированные системы управления технологическими процессами — АСУ ТП.

Класс В — системы (подсистемы) подготовки и учета производственной деятельности предприятия. К таким системам относят те, которые находятся на уровне технологического процесса, но не связаны с технологией напрямую. Они предназначены для выполнения учетных задач, для сбора и предварительной подготовки данных, поступающих из систем класса А, и передачу их в систему класса С. Они могут выполнять задачи управления ресурсами в рамках технологического процесса, планирования и контроля последовательности операций в нем, управления каче-

ством продукции, техническим обслуживанием и ремонтом, хранением материалов и продукции.

В качестве примера систем класса В можно привести системы управления производством — MES (Manufacturing Execution Systems), системы планирования потребностей в материалах — MRP (Material Requirements Planning), системы планирования ресурсов производства — MRP II (Manufacturing Resource Planning), системы планирования производственных мощностей — CRP (C Resource Planning), автоматизированные системы проектирования (САПР) — CAD (Computing Aided Design), автоматизированные системы поддержки производства — CAM (Computing Aided Manufacturing), автоматизированные системы инженерного проектирования (САПР) — CAE (Computing Aided Engineering), автоматизированные системы управления данными — PDM (Product Data Management), системы управления взаимоотношениями с клиентами — SRM (Customer Relationship Management), разные учетные системы.

Класс С — системы (подсистемы) планирования и анализа производственной деятельности предприятия. На основе информации, поступающей из систем класса В, эти системы выполняют анализ деятельности предприятия. Они решают задачи планирования его деятельности, регулирования параметров работы предприятия, планирования и распределения ресурсов. В системах класса С выполняется подготовка производственных заданий и контроль их исполнения. Обработка информации идет в интерактивном режиме. Эти системы оказывают влияние на деятельность предприятия в целом.

В качестве примера систем класса С можно привести системы планирования ресурсов предприятия — ERP (Enterprise Resource Planning), системы интеллектуального планирования — IRP (Intelligent Resource Planning), автоматизированные системы управления предприятиями — АСУП.

ИИС — интегрированные информационные системы. Создаются для автоматизации всех как организационных, так и производственных функций организации (предприятия). Это, как правило, большие АСУ — системы человек—машина, обеспечивающие эффективное функционирование объекта, в которых сбор и переработка информации необходимы для реализации функций организационного и технологического управления. Цель функционирования такой системы — оптимизация работы объекта по

заданным критериям управления (технико-экономическим и/или технологическим показателям).

Например, интегрированные финансовые системы (ИФС). Такие системы, как наиболее распространенные в настоящее время «БЭСТ», «Галактика», «Парус», являются модульными, т. е. состоят из отдельных блоков учета. Они считаются интегрированными в области финансов и учета, так как существует двусторонняя связь между всеми модулями системы, каждый из которых выполняет свои функции учета. Это следующие модули: «Главная книга», «Основные средства», «Управление запасами», «Расчеты с поставщиками и заказчиками», «Касса и банк», «Зарплата», «Кадры» и т. д. Благодаря принципу интегрированности отсутствуют проблемы двойного ввода документов при обработке.

Интегрированная система имеет такие общие свойства, как общий документированный учет, степень интеграции, удобный интерфейс.

А системы «бухгалтерия-офис» предназначены не только для бухгалтера, но и для руководителя, менеджера и другого управленческого персонала. Особое внимание в этих системах уделяется возможности эффективного управления предприятием и получению прибыли. Бухгалтерский учет направлен на решение задач по обобщению показателей деятельности предприятия. В то же время с помощью оперативно-управленческого учета решаются ежедневные задачи. Одна форма учета тесно связана с другой.

В качестве примера систем «бухгалтерия-офис» могут служить Baan, R3, People soft, «Квестор» и др.

Классификация АИС по *масштабу*: одиночные, групповые, корпоративные.

Одиночные АИС реализуются, как правило, на автономном персональном компьютере (сеть не используется). Такая система может содержать несколько простых приложений, связанных общим информационным фондом. Она рассчитана на работу одного пользователя или группы пользователей, разделяющих по времени одно рабочее место. Подобные приложения создаются с помощью так называемых настольных, или локальных, систем управления базами данных (СУБД). Среди локальных СУБД наиболее известными являются Clarion, Clipper, FoxPro, Paradox, dBase и Microsoft Access.

Групповые АИС ориентированы на коллективное использование информации членами рабочей группы и чаще всего строятся

на базе локальной вычислительной сети. При разработке таких приложений используются серверы баз данных (называемые также SQL-серверами) для рабочих групп. Существует довольно большое количество различных SQL-серверов, как коммерческих, так и свободно распространяемых. Среди них наиболее известны такие серверы баз данных, как Oracle, DB2, Microsoft SQL Server, InterBase, Sybase, Informix.

Корпоративные АИС являются развитием систем для рабочих групп. Они ориентированы на крупные компании и могут поддерживать территориально разнесенные узлы или сети. В основном такие системы имеют иерархическую структуру нескольких уровней. Для них характерна архитектура клиент/сервер со специализацией серверов или же многоуровневая архитектура. При разработке корпоративных систем могут использоваться те же серверы баз данных, что и при разработке групповых АИС. Однако в крупных АИС наибольшее распространение получили серверы Oracle, DB2 и Microsoft SQL Server.

Для групповых и корпоративных систем существенно повышаются требования к надежности функционирования и сохранности данных. Эти свойства обеспечиваются поддержкой целостности данных, ссылок и транзакций в серверах баз данных.

Классификация групповых и корпоративных АИС *по способу организации*: на основе архитектуры файл-сервер, архитектуры клиент/сервер, многоуровневой архитектуры, технологий Интернет/интранет.

Классификация АИС *по видам выполняемых операций*: обработки транзакций, аналитической обработки, поисковые и справочные системы, управления документооборотом.

Системы обработки транзакций, в свою очередь, по оперативности обработки данных разделяются на пакетные и оперативные АИС. В АИС организационного управления преобладает режим оперативной обработки транзакций (OnLine Transaction Processing, OLTP) для отражения актуального состояния предметной области в любой момент времени, а пакетная обработка занимает весьма ограниченную часть.

Для систем OLTP характерен регулярный (возможно, интенсивный) поток довольно простых транзакций, играющих роль заказов, платежей, запросов и т. п. Важными требованиями для них являются высокая производительность обработки транзакций и гарантированная доставка информации при удаленном доступе к БД по телекоммуникациям.

Системы аналитической обработки (поддержки принятия решений — Decision Support System, DSS) представляют собой другой тип АИС, в которых с помощью довольно сложных запросов производится отбор и анализ данных в различных разрезах: временных, географических, по другим показателям. Делятся *на системы оперативной аналитической обработки и обработки знаний в экспертных системах.*

Системы управления документооборотом делятся на следующие системы: документальные, автоматизированного делопроизводства, управления документооборотом.

Приводимые классификации в достаточной степени условны. Крупные АИС очень часто обладают признаками всех перечисленных выше классов. Кроме того, корпоративные АИС обычно состоят из ряда подсистем, относящихся к различным сферам применения.

Контрольные вопросы

1. Что значит «классифицировать АИС»? Приведите основания классификации.
2. По каким принципам происходит деление АИС по масштабу?
3. Какова классификация групповых и корпоративных АИС по способу организации?
4. Как классифицировать АИС по видам выполняемых операций?
5. Какие АИС относятся к экономическим?
6. Что понимают под одноуровневой и многоуровневой многоцелевыми системами?
7. Какие подсистемы должны входить в состав отраслевой информационной ОАСУ?

10.1.1. Система «КонсультантПлюс»

С 90-х годов XX в. получает признание и начинает успешно развиваться правовая информатика — отрасль науки, изучающая информацию, информационные процессы и информационные системы в праве (или в правовой системе). На рынке справочно-правовых систем представлены «КонсультантПлюс», «Гарант», «Кодекс». Эти системы — основной инструмент принятия

решений по правовым вопросам для бухгалтеров, юристов, руководителей и других специалистов.

В правовой системе объект рассмотрения — в первую очередь информация, которая находится в гражданском, административном или ином общественном обороте и в связи с которой потому и возникают общественные отношения, подлежащие регулированию правом. Рассмотрим одну из таких систем.

Автоматизированная справочная правовая система «КонсультантПлюс», разработана НПО «Вычислительная математика и информатика». Система является мощным средством оперативного получения полной и достоверной нормативной информации. Информационный банк (ИБ) системы пополняется информацией, непрерывно поступающей от центральных органов государственной власти и управления, а также других организаций, выпускающих нормативные документы правового характера. Программная оболочка системы предоставляет удобство работы для пользователя и осваивается достаточно просто (рис. 10.4). Система имеет следующие разделы:

- информационно-правовые комплексы «КонсультантПлюс»;
- раздел «КонсультантПлюс: Законодательство»;
- раздел «КонсультантПлюс: Международные правовые акты»;
- раздел «КонсультантПлюс: Правовые акты по здравоохранению»;
- раздел «КонсультантПлюс: Технические нормы и правила»;
- раздел «КонсультантПлюс: Формы документов»;
- раздел «КонсультантПлюс: Комментарии законодательства» (постатейные комментарии и книги, юридическая пресса);
- раздел «КонсультантПлюс: Судебная Практика»;
- раздел «КонсультантПлюс: Финансовые консультации»;
- раздел «КонсультантПлюс: Законопроекты»;
- ИПК «КонсультантПлюс: Эксперт»;
- СПС «КонсультантПлюс: Версия Проф»;
- СПС «КонсультантПлюс: Эксперт-приложение»;
- СПС «КонсультантПлюс: Российское законодательство»;
- СПС «КонсультантПлюс: Налоги и бухучет»;
- СПС «КонсультантПлюс: Документы СССР»;
- СПС «КонсультантПлюс: Регионы»;
- СПС «КонсультантПлюс: Сводное Региональное законодательство»;
- ИБ «КонсультантПлюс: Московский выпуск»;

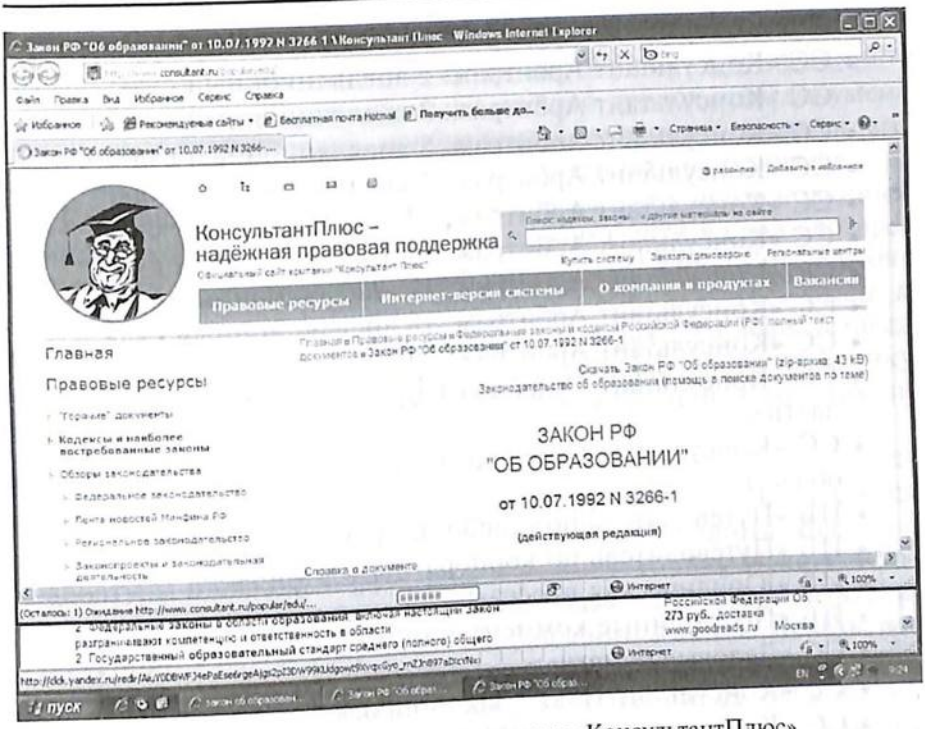


Рис. 10.4. Результат поиска в системе «КонсультантПлюс»

- СПС «КонсультантПлюс: Санкт-Петербург и Ленинградская область»;
- ИПК «Московский Бухгалтер»;
- СС «Консультант Бухгалтер: Версия Проф»;
- ИБ «КонсультантПлюс: Нормативные документы»;
- СС «Консультант Финансист»;
- ИБ «Бухгалтерская пресса и книги»;
- ИБ «Путеводитель по налогам»;
- ИБ «Путеводитель по сделкам»;
- ИБ «Судебная практика для Бухгалтера»;
- СС «Консультант Бухгалтер: Корреспонденция счетов»;
- СС «Консультант Судебная Практика: Подборки судебных решений»;
- СС «Консультант Судебная Практика: Решения высших судов»;
- СС «Консультант Арбитраж: ФАС всех округов»;
- СС «Консультант Арбитраж: Налоговые споры»;

- СС «Консультант Арбитраж: 1 апелляционный суд»;
- СС «Консультант Арбитраж: 2 апелляционный суд»;
- СС «Консультант Арбитраж: 3 апелляционный суд»;
- СС «Консультант Арбитраж: 5 апелляционный суд»;
- СС «Консультант Арбитраж: 7 апелляционный суд»;
- СС «Консультант Арбитраж: 13 апелляционный суд»;
- СС «Консультант Арбитраж: 14 апелляционный суд»;
- СС «Консультант Арбитраж: 17 апелляционный суд»;
- СС «Консультант Арбитраж: 18 апелляционный суд»;
- СС «Консультант Арбитраж: 19 апелляционный суд»;
- СС «Консультант Судебная Практика: Суды Москвы и области»;
- СС «Консультант Судебная Практика: Суды Свердловской области»;
- ИБ «Путеводитель по судебной практике (ГК РФ)»;
- ИБ «Путеводитель по корпоративным спорам»;
- ИБ «Юридическая пресса»;
- ИБ «Постатейные комментарии и книги»;
- СС «Деловые бумаги»;
- СС «КонсультантПлюс: Законопроекты»;
- СС «КонсультантПлюс: Международное право»;
- СПС «КонсультантПлюс МедицинаФармацевтика»;
- СС «КонсультантПлюс: Строительство».

Система предназначена для быстрого поиска и получения полной и достоверной нормативной информации. Она позволяет осуществлять поиск списка необходимых документов, текста нужного документа, дополнительной информации о документах, позволяет редактировать текст документа, заносить его в файл и распечатывать на принтере. Она имеет и другие функциональные возможности (занесение документов в отдельные папки, установление закладок в тексте и др.).

Для работы с ИБ системы следует прежде всего освоить основные *функциональные клавиши*, изучив руководство по работе с системой.

Следует заполнить карточку реквизитов, внося в нее необходимые реквизиты искомого документа. Этот процесс упрощен за счет использования встроенных словарей для каждого из реквизитов. Из этого словаря следует выбрать несколько подходящих значений, соединив их логическими условиями И, ИЛИ, КРОМЕ. После заполнения любого поискового поля в карточке сис-

тема сразу определяет количество документов, удовлетворяющих введенным данным. После нажатия соответствующей функциональной клавиши можно вывести полный список этих документов, а далее, определив необходимый документ, вывести его текст на экран, занести в файл или распечатать на принтере.

Карточка реквизитов состоит из нескольких поименованных поисковых полей. С каждого поля можно выйти в свой словарь, состоящий из реквизитов, которыми следует заполнить карточку. Для заполнения карточки нужно подвести указатель мыши к выбранному полю и, щелкнув мышью, вызвать словарь. Занеся в карточку необходимые поисковые признаки и нажав кнопку **Поиск**, выводим на экран список документов или необходимый документ.

Для быстрого поиска нужного документа в любом из поисковых полей нужно нажать кнопку **Найти**. На экране появится окно **Найти** со строкой для ввода нужного фрагмента. Введем с клавиатуры поисковый фрагмент и нажмем кнопку **Искать**. Курсор установится на первом вхождении заданного фрагмента. Для повторения поиска в нужном направлении кнопка **Искать** используется столько раз, сколько необходимо.

Поиск документа можно осуществить из списка, если он небольшой. Находим название документа, нажимаем клавишу **Enter** и осуществляем просмотр текста документа.

Пример 10.1. Найти принятые начиная с 1997 г. документы, в которых говорится о государственных пошлинах (поиск рубрики в тематическом рубрикаторе).

Последовательность работы:

1. Откройте окно системы «КонсультантПлюс», щелкнув мышью по ее пиктограмме на рабочем столе.

2. Выберите в главном меню систему **К+Проф** и откройте ее окно.

3. Для работы в системе изучите основные функциональные клавиши.

4. Откройте карточку реквизитов и изучите все поисковые поля (**Тематика**, **Вид документа**, **Принявший орган**, **Дата принятия**, **Регистрационный номер**, **Статус документа**, **Текст**, **Ключевые слова** и др.) и встроенные словари для каждого из них.

5. Используя словари поисковых полей, научитесь проводить поиск документов по реквизитам. Поиск проводится по тематике. Очистите карточку реквизитов при помощи кнопки **Удалить все**.

Дважды щелкните по полю **Тематика** для входа в словарь — многоуровневый рубрикатор. Если слева от рубрики стоит символ +, то, щелкнув по нему, можно увидеть следующий уровень, т. е. подрубрики. Нажмите кнопку **Найти** в нижней части окна **Тематика**. В появившемся поисковом окне наберите слово «пошлин*» (символ * предполагает любое окончание). Нажмите кнопку **Искать**. Рубрикатор раскроется, и курсор установится на рубрике «Государственные пошлины». Нажмите кнопку **Выбрать** для занесения рубрики в карточку реквизитов. Нажав кнопку **Поиск**, сформируйте список документов, где говорится о госпошлинах.

Пример 10.2.

1. *Исходные данные.*

Документ «Постановление о порядке исчисления среднего заработка в 1998 году».

2. *Последовательность работы.*

Выбор показателей для поиска документов: выбираем в карточке реквизитов поисковые поля **Вид документа**, **Название документа**, **Дата принятия**.

3. *Проводим работу по поиску документов.*

Очищаем карточку реквизитов.

Для поиска документа «Постановление о порядке исчисления среднего заработка в 1998 году» выберем в карточке реквизитов поисковые поля **Вид документа**, **Название документа**, **Дата принятия**. В словарях указанных полей выбираем соответственно реквизиты ПОСТАНОВЛЕНИЕ, СРЕДН* рядом ЗАРАБОТ*, с 1 января 1998 г. по 31 декабря 1998 г. Осуществляем поиск и просмотр документа. Для занесения его в файл нажимаем функциональную клавишу **F2**, а для распечатки на принтере — клавиши **Ctrl+F2**.

Контрольные вопросы

1. Каковы назначение и функции системы КонсультантПлюс?
2. Каковы функции карточки реквизитов?
3. Как осуществляется поиск списка документов?
4. Как осуществляется поиск документов?
5. Приведите пример заполнения карточки реквизитов и поиска документа в системе КонсультантПлюс.

10.2. Направления развития АИС

Перспективы развития современных АИС базируются на следующих важнейших направлениях:

1. Эволюция информационно-поисковых систем по цепочке «ИПС — интегральная информационная система — корпоративная информационная система — информационный портал».
2. Совершенствование взаимосвязи АИС с автоматизированными системами управления организационного (АСУП) и технологического типа (АСУТП).
3. Интеграция неоднородных информационных ресурсов.
4. Активное использование объектных технологий.
5. Развитие архитектуры распределенных систем.
6. Мобильные АИС.
7. Поддержка метаданных.
8. Семантическая обработка информационных ресурсов.
9. Управление потоками данных.
10. Совместное использование информационных технологий.
11. Рост масштабов АИС.
12. Глобализация АИС.
13. Конвергенция технологий.
14. Развитие стандартов информационных технологий.
15. Автоматизированная разработка АИС.
16. Компонентное программирование.

Эволюция информационно-поисковых систем по цепочке «ИПС — интегральная информационная система — корпоративная информационная система — информационный портал». ИПС — это базовый класс для АИС. В современных АИПС стоит задача их интеллектуализации, т. е. развития процедур автоматического индексирования полнотекстовых документов на основе «интеллектуальных тезаурусов», языка высокого уровня и автоматического синтаксического анализа текста. Появляется возможность одновременной обработки миллионов полнотекстовых электронных документов, проиндексированных для поиска на нескольких уровнях. Например, поисковая система «Галактика-Зум». Эта система является уже лингвистическим процессором, работающим с большими и сверхбольшими массивами текстовой информации и решающим информационно-аналитические задачи.

Развитие компьютерной лингвистики и, в частности, методы, основанные на примерах и аналогии, позволяют совершенствовать автоматическую обработку текстовой информации. Раз-

работаны системы фразеологического машинного перевода политематических текстов RETRANS (русский—английский), ERTRANS (английский—русский). По естественным и техническим наукам создан политематический тезаурус ключевых слов и словосочетаний.

Среди АИПС особо можно выделить интегральные информационные системы, разработанные в ВИНИТИ (Всероссийский институт научно-технической информации). Разработки института, которые начаты в 1965—1969 гг., были восприняты затем многими отраслевыми и межотраслевыми системами, как ИИС, так и АСУ (например, ОАСУ «Электроника»). На современном этапе разработки используются в системах управления документооборотом (Docflow) и системах управления заданиями (Docroute).

Последние разработки — информационные порталы. В них также прослеживаются идеи и технологии ИИС, как и в получивших распространение корпоративных информационных системах.

Информационный портал — система массового обслуживания с одним входом (входным каналом связи) или сложная система, выполняющая одну или несколько информационно-технологических функций.

Например, для региональной аналитической службы администрации Новгородской области исследовалась технология одноканального ввода гетерогенных документов в сеть функциональных серверов, исполняющих роль электронного распределенного хранилища для нескольких АИС: «Письма граждан», «Инвестиционные документы и проекты», «Нормативные акты федерального и регионального уровней», «Статистика», «Научно-методическая литература», «СМИ», «Приказы и распоряжения Администрации», «Текущие мероприятия», «Аналитика». Создается внутренняя локальная сеть, объединенная через средства информационной защиты с внешними сетями, в частности с Интернетом. Все серверы и сети управляются с корпоративного сервера. Такую технологию можно рассматривать как технологию работы информационного портала. Он управляет электронными ресурсами нескольких АИС, созданных на реляционных или объектных СУБД, с общим входом, контролируемым администратором по информационно-аналитическим ресурсам.

Еще пример. Корпоративный портал DDB нового поколения, разработанный компанией Exteria по технологии MS Digital

Dashboard 2.01, из стандартных многократно используемых компонентов на языке XML. Портал объединяет информацию и средства управления агентств и изданий: «Интерфакс», «Прайм-ТАСС», «Алгоритм», DHL, «Известия», Price.ru и др.

Интернет-порталы на программах систем управления документами (DOCSOpen) будут использованы в региональных аналитических службах административного управления муниципальных образований и в крупных информационно-аналитических центрах — генераторах баз данных.

Совершенствование взаимосвязи АИС с автоматизированными системами управления организационного (АСУП) и технологического типа (АСУТП) начинает заинтересовывать как промышленные предприятия, так и информационные центры, библиотеки, издательства, учебные заведения. Взаимосвязь указанных систем по информационно-лингвистическому и программному обеспечению помогает на практике при автоматическом поиске информации. Так, SQL-сервер можно отнести и к поисковым системам, и к системам управления электронными ресурсами, и к технологии реляционных БД. В последнее время возникла идея распределенного управления народным хозяйством с помощью автоматизированных систем (АС), локально обрабатывающих информацию. Реализация этой идеи требует создания АРМ на базе профессиональных ПЭВМ для каждого уровня управления.

Принципы создания должны быть общими: системность, гибкость, надежность, эффективность. Структура таких АРМ должна определяться их функциональностью. Подсистемы и элементы их должны строиться по модульному принципу, чтобы быть адаптивными, т. е. приспосабливаться к перестройке. Необходимо, чтобы они отвечали требованиям стандартов. Неполадки в системе должны быть легко устранимы, работоспособность должна быстро восстанавливаться.

Распределенные АС требуют специальных каналов передачи данных. Для передачи данных используют следующие постоянно совершенствующиеся виды сетей:

- LAN — Local Area Network. Используются для связи и передачи данных между удаленными автоматизированными системами в пределах здания;
- MAN — Metropolitan Area Network. Используются для связи и передачи данных между удаленными АС в пределах города;

- WAN — Wide Area Network. Используются для связи и передачи данных между удаленными АС в пределах страны или региона.

В дальнейшем будут развиваться так называемые сквозные системы — всеобъемлющий набор средств для автоматизации процессов и технологической подготовки производства, а также различных объектов промышленности. Намечают совмещение трех основных типов САПР (CAD, CAM, CAE). Это позволит в масштабе предприятия логически связывать всю информацию об изделии, обеспечивать быструю обработку и доступ к ней пользователей, работающих в разнородных системах.

Таким образом, административные системы должны разрабатываться на базе многозадачных ОС и локальных, а в перспективе и глобальной сетевых сред. Важнейшим вопросом, решаемым ПО таких АИС, должен стать вопрос контроля доступа к данным. Уже сейчас персональная вычислительная техника и ПО доступны. Для АИС, функционирующих в персональной среде, неизбежен скорый переход в среду многопользовательскую. Учитывая тенденции к интеграции, аппаратные и программные средства АИС следует выбирать такие, для которых гарантированы программы развития и возможность интеграции как в однородные, так и в неоднородные среды.

Активное использование объектных технологий. В разработках АИС прочные позиции заняли объектные технологии. Их расширению способствует создание развитой объектной инфраструктуры. Важны общие элементы объектной инфраструктуры, независимые от класса АИС, а также элементы, ориентированные на отдельные классы систем, — системы баз данных, сетевые, текстовые системы. Объектное направление в области АИС хорошо оснащено инструментальными средствами CASE, основанными на методах объектного анализа и проектирования и использующими стандартизованный язык UML (Unified Modeling Language) для представления метаданных.

Интеграция неоднородных информационных ресурсов. Благодаря активным разработкам АИС многие организации стали обладателями коллекций информационных ресурсов разной природы, каждая из которых поддерживается собственными программными средствами и своим интерфейсом. Пользователю нужен единый интерфейс для доступа ко всем информационным ресурсам. Под интеграцией информационных ресурсов понимается обеспечение пользователям доступа к нескольким источни-

кам информационных ресурсов в терминах единого представления, исключающего избыточность информации на логическом или семантическом уровне. Некоторые относительно простые возможности интеграции обеспечиваются программными продуктами. Более сложные проблемы семантической интеграции пока еще являются предметом изучения.

Развивается принцип «однократного (одноразового) ввода информации и многократного (многоцелевого или многофункционального) ее использования». Принцип предполагает в ИИС один раз проводить аналитическую обработку документов (реферирование или иное сжатие информации), а далее использовать многократно при различном информационно-аналитическом обслуживании. Принцип однократного ввода получил развитие в системах управления документами компании «Весть-Метатехнология» DOCSOpen, которые могут работать с документами разной структуры (гетерогенными документами). Они могут быть законодательными, научно-техническими, проектно-конструкторскими, нормативными (стандарты), бухгалтерскими, банковскими и др.

Другой принцип — «принцип эмерджентности». Рассматривается как новое качество системы, возникшее с возможностью извлекать и синтезировать новые сведения, в явном виде не введенные в электронный архив (в литературе используются разные термины — Data Mining, Knowledge Discovery и т. д.).

Развитие архитектуры распределенных систем. В многочисленных корпоративных АИС используются распределенные базы данных. Отработаны методы распределения данных и управления ими, архитектурные подходы, обеспечивающие масштабируемость систем, реализующие принципы многозвенной архитектуры клиент/сервер, а также архитектуры промежуточного слоя. Начинают применяться на практике мобильные архитектуры. Это относится как к системам баз данных, так и к веб-приложениям. Возрождается подход к построению распределенных систем, основанный на одноранговой архитектуре (Peer-to-Peer), при котором, в отличие от доминирующей сегодня в распределенных системах архитектуры клиент/сервер, роли взаимодействующих сторон в сети не фиксируются. Они назначаются в зависимости от ситуации в сети, от загруженности ее узлов.

Мобильные АИС. В связи с интенсивным развитием коммуникационных технологий активно развиваются мобильные АИС. Разработаны технические средства и программное обеспечение

для их создания. Благодаря этому стали развиваться мобильные системы баз данных. Многие научные коллективы проводят исследования специфических особенностей таких систем, создают разнообразные их прототипы. Важным инструментом для разработки мобильного программного обеспечения стали технологии Java.

Создан стандарт протокола беспроводного доступа приложений к Сети (Wireless Application Protocol, WAP), который уже поддерживается некоторыми моделями сотовых телефонов. На основе WAP и языка XML консорциум W3C разработал язык разметки для беспроводных коммуникаций WML (Wireless Markup Language).

Поддержка метаданных. В разработках АИС больше внимания стали уделять метаданным, а именно стандартизации представления метаданных и обеспечению их поддержки в системе. Активно ведутся разработки стандартов, ориентированных на различные информационные технологии. Наиболее значимые стандарты: из первых — язык описания данных CODASYL для БД сетевой структуры; из последних — стандарт языка запросов SQL для реляционных БД; компонент стандарта объектных БД ODMG, международный стандарт IRDS (Information Resource Dictionary Systems), описывающий системы для создания и поддержки справочников информационных ресурсов организации, и др. Новая платформа XML для Сети также включает стандарты представления метаданных. Разработан стандарт языка UML (Unified Modeling Language), обеспечивающий представление метаданных инструментов CASE для визуального объектного анализа и проектирования (разработан OMG). Язык поддерживается во многих программных продуктах CASE.

Семантическая обработка информационных ресурсов. Еще в 1970—1980-е годы были созданы различные исследовательские прототипы систем, основанных на знаниях, поддерживающих семантические модели данных, а также информационно-поисковых систем, в которых в качестве языков запросов использовались естественные языки.

В последние годы активно велись работы по семантическому текстовому поиску в нашей стране, консорциумом W3C, другими центрами в США и Европе. Активно проводятся работы по созданию семантической паутины. В то время как действующая реализация Сети предусматривает интерпретацию информационных ресурсов человеком, семантическая паутина позволит

создавать приложения с компьютерной их интерпретацией. Она будет располагать также средствами логического вывода.

Управление потоками данных. Это одно из новых направлений в области АИС, связанное с обработкой данных сетевого трафика, данных, порождаемых различного рода датчиками, потоков сообщений электронной почты и т. п. Стали создаваться предназначенные для этой цели инструментальные средства.

Важный элемент информационной инфраструктуры — документ рассматривается как часть информационного потока, как элемент административных процессов управления, как объект периодического анализа, как единица хранения в АИС или архивах и библиотеках, как финансовый или исторический документ. Документ может быть электронным, он участвует в документообороте, делопроизводстве, имеет жизненный цикл. Документация может быть организационно-распорядительная, проектная, научно-техническая. Придается важнейшее значение управлению документооборотом, созданию электронных архивов (в том числе корпоративных), офисов, канцелярий, маршрутизации документов и заданий, сети электронных библиотек. Power DOC — комплекс автоматизации делопроизводства на русском языке, Work Route II — отечественная система автоматизации деловых процессов. Проводятся исследования процедур извлечения данных, знаний из традиционных и электронных массивов.

От качества средств регистрации и обработки первичных данных, систем их сбора и передачи зависит и достоверность обрабатываемых данных. Универсальность этих устройств может быть получена при условии выработки стандартных протоколов сопряжения ЭВМ при их объединении в сети для всех типов ЭВМ и всех уровней их объединения. Протоколы должны допускать передачу информации между абонентами различных типов (человек, ЭВМ, техническое средство и т. п.). Сами системы передачи данных должны иметь развитые методы коммутации высокоскоростных каналов, цифровую передачу по кабельным, радиорелейным, спутниковым и волоконно-оптическим линиям с соответствующими концентраторами данных и аппаратурой ввода.

Совместное использование информационных технологий. В последние годы стали появляться инструментальные средства и крупные АИС, в которых совместно используются различные информационные технологии из области баз данных, текстовых систем и Сети. Создан ряд коммерческих СУБД, которые предоставляют возможность текстового поиска. Простейшие возможно-

сти контекстного поиска обеспечивают популярные веб-браузеры. Поисковые машины Сети используют реализованную в этой среде технологию доступа к информационным ресурсам вместе с технологиями текстового поиска. Широко развивается технология мультимедиа, объединяющая использование разных информационных технологий.

Рост масштабов АИС. Масштаб систем зависит не только от объема поддерживаемых информационных ресурсов, но и от числа их пользователей. Появились системы очень больших БД, поддерживающие многие гигабайты и даже петабайты данных, системы текстового поиска с очень большими коллекциями документов. Объем информационных ресурсов Сети в настоящее время исчисляется многими миллионами страниц. Корпоративные системы баз данных насчитывают тысячи пользователей. На порядок больше пользователей имеют некоторые информационные сервисы Сети. Количество таких крупных систем продолжает расти.

Глобализация АИС. Глобализация АИС имеет две стороны — обеспечение глобального доступа пользователей к системе и интеграцию информационных ресурсов, распределенных в глобальной сети. Указанные стороны глобализации АИС воплощены в уникальной глобальной АИС Сети. Многочисленные глобальные системы создаются в настоящее время как веб-приложения для электронного бизнеса, для поддержки научной кооперации различных коллективов ученых во многих областях знаний в международном и национальном масштабе, в библиотечном деле и в других сферах.

Конвергенция технологий. Имеет место взаимопроникновение идей, заимствование подходов и техники из смежных областей информационных технологий.

В системах текстового поиска используются заимствованные из технологий БД методы прямого доступа к информационным ресурсам на основе техники индексирования. Веб-технологии используют методы текстового поиска, многие ключевые концепции и подходы к управлению данными, созданные в области баз данных. В свою очередь, в технологиях БД зарождается новый класс систем БД, поддерживающий коллекции XML-документов. Появились коммерческие XML-ориентированные СУБД.

Развитие стандартов информационных технологий. Последнее десятилетие эта деятельность осуществляется не только силами официальных органов стандартизации, но и многочисленными

специально для этих целей учрежденными индустриальными консорциумами. Благодаря созданию стандартов обеспечивается переносимость приложений и информационных ресурсов между различными программно-аппаратными платформами, интероперабельность программных продуктов различных поставщиков и созданных на их основе приложений, повторное использование ресурсов, в частности метаданных и программных компонентов приложений. Все большее признание специалистов получают стандарты системного проектирования, обеспечивающие эффективное управление жизненным циклом создаваемой системы, отсутствие упущений в процессе разработки, высокое ее качество.

Автоматизированная разработка АИС. Крупное достижение технологий современных АИС состоит в создании методов их анализа и проектирования, которые в течение двух-трех десятилетий прошли испытания на практике. На их основе разработаны инструментальные средства CASE, которые поставляются многими компаниями — разработчиками ПО. Такие технологии широко применяются прежде всего для создания систем баз данных. Важное место в этой области принадлежит методам объектного анализа и проектирования.

Компонентное программирование (КП). Современный этап технологий программирования представляет собой компонентное программирование (COM Object Model) как тенденцию развития ООП (технологии COM, COM+, .Net (компании Microsoft), CORBA, Jawa и др.). Одна из целей КП — создание технологии для разработки (программирования) распределенных систем. Распределенность, независимость — некоторые из ключевых слов, характеристик КП в целом. Компонент — это хранилище (в виде DLL- или EXE-файлов) для одного или нескольких классов.

Компонентные технологии позволяют формировать сложные распределенные приложения, некоторые части которых выполняют в различных узлах локальной или глобальной сети. Основная идея КП — распространение классов в бинарном виде (т. е. не в виде исходного кода) и предоставление доступа методам класса через строго определенные интерфейсы. Это позволяет снять проблему несовместимости компиляторов и обеспечивать смену версий классов без перекомпиляции использующих их приложений.

В настоящее время повышается культура проектирования и реализации крупных АИС, основанных на технологиях БД. В ус-

ловиях рыночной экономики уделяется серьезное внимание управлению проектами систем, не только технологическим, но и экономическим их аспектам. Для этого развиваются необходимые методы и создаются инструментальные средства.

Контрольные вопросы

1. В чем заключается объектный подход в построении АИС?
2. Что такое CASE-технологии?
3. В чем заключается суть интеграции информационных ресурсов?
4. Какова архитектура распределенных систем?
5. Что такое мобильные АИС?
6. Чем характеризуется совместное использование информационных технологий?
7. Каковы основные признаки глобализации АИС?
8. Какова тенденция развития стандартов информационных технологий?

Заключение

Современный этап развития информатики, авутоматизированных информационных технологий и систем характеризуется необходимостью решения сложных задач в различных сферах человеческой деятельности. Автоматизация не только рутинных, но и высокоинтеллектуальных операций является уже повседневной задачей.

Создание ЭВМ высокой производительности с использованием принципа распараллеливания операций позволяет автоматизировать многие сложные процессы. Уже в настоящее время производительность универсальных ЭВМ составляет $10^5...10^7$ операций в секунду, а в будущем ожидается производительность 10^{95}. Такие сверхбыстродействующие универсальные ЭВМ будут иметь в своем составе 20—30 процессоров с производительностью каждого процессора $3 \cdot 10^6...6 \cdot 10^6$. Будут создаваться специализированные процессоры, например для научных вычислений, с производительностью $3 \cdot 10^8...6 \cdot 10^8$ операций в секунду, матричные процессоры с производительностью до 10^7 операций в секунду, процессоры БД, ориентированные на оптимальную обработку запросов к локальным и распределенным базам данных, процессоры передачи данных, обеспечивающие функции коммутации пакетов, работу по цифровым каналам связи. Создаются микро-ЭВМ и супермикро-ЭВМ, позволяющие повысить надежность вычислительных комплексов в 5—10 раз. Продолжается развитие систем ввода-вывода в направлении разработки синтезаторов-анализаторов речи, лингвистических процессоров. Все эти устройства предназначены для обеспечения более удобного общения пользователя с ЭВМ, в частности, на естественном языке. Синтезаторы должны поддерживать режимы речевых команд; звуковой мнемоники; диалога.

Уже не за горами использование биокомпьютеров. Первый биологический компьютер, использовавший ДНК в качестве но-

сителя информации, был создан в 1994 г. Группе ученых из израильского университета удалось создать пробный образец биокомпьютера, который представляет собой обычную стеклянную пробирку, заполненную молекулами ДНК, РНК и различными ферментами. Принцип работы такого компьютера заключается в том, что информация — молекулы ДНК — обрабатывается аппаратной частью — ферментом. Проще говоря, между ними происходит химическая реакция, продуктом которой и является результат обработки информации.

Именно эти компоненты и выступают в роли устройств ввода/вывода, вычислительного блока и программного обеспечения. Миллилитр ДНК содержит больше информации, чем триллион CB-H. Столовая ложка «компьютерного бульона» по производительности в миллионы раз превосходит используемые нами персональные компьютеры. Однако применение такого «бульона» сильно отличается от задач, которые стоят перед обычными вычислительными машинами.

Исследования проводились в Вейцмановском институте (Weizmann Institute) в Израиле под руководством профессора Эхуда Шапиро (Ehud Shapiro), который так описал свое изобретение: «Нам удалось создать настоящий биокомпьютер. Причем размеры его настолько малы, что в одну пробирку можно поместить до триллиона вычислительных модулей, суммарная производительность которых составит один миллиард операций в секунду». Точность вычислений составляет 99,8 %, правда, все результаты обработки информации «выдаются» только в виде двух вариантов ответа: «истина» или «ложь». Причем сам результат вычислений тоже представляет собой биоматериал: это молекула, образовавшаяся в результате взаимодействия ДНК, РНК и ферментов (кстати, именно ферменты выступают в роли вычислительных модулей, тогда как в молекулах ДНК зашифрованы исходные данные и программное обеспечение). И хотя исследователям уже удалось создать опытный образец биокомпьютера, первые полноценные устройства, по мнению ученых, появятся не раньше чем через несколько десятилетий. Так что владельцам ПК пока что не стоит опасаться того, что в скором времени их машины превратятся в груду никому не нужного железа.

Разработчик полагает, что в перспективе такие биологические компьютеры смогут работать «внутриклеточными докторами», которые способны реагировать на возникающие аномалии согласно заложенным в них программам — вырабатывая лекар-

ства. За созданием компьютеров на ДНК не успевает даже научная фантастика.

Наука быстро развивается. Необходимо, чтобы практика от нее не отставала. А это зависит от того, какой квалификации специалисты придут на производство. Подготовка инженеров и техников, владеющих современными знаниями в области информатики, автоматизированных информационных технологий и систем, — первоочередная задача. Эта книга должна помочь студентам стать такими специалистами.

Литература

1. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. М., 2000.

2. Единая система программной документации (ЕСПД). М., 2000.

3. Законы РФ: «О стандартизации» от 10.06.1993 (последняя редакция от 25.07.2002); «О сертификации продукции и услуг» от 27.04.1993 (последние изменения от 10.01.2008, ч. 4 ГК РФ); «Об информации, информационных технологиях и о защите информации» от 27.07.2006; «О правовой охране для электронных вычислительных машин и баз данных» от 23.09.1992 и др.

4. ГОСТ Р ИСО/МЭК 9126—93. Информационная технология. Оценка программной продукции. Характеристики качества и руководство по их применению. Государственный стандарт Российской Федерации. Издание официальное. М., 1994.

5. *Гвоздева В. А.* Введение в специальность программиста: учебник. М., 2007.

6. *Гвоздева В. А., Лаврентьева И. Ю.* Основы построения автоматизированных информационных систем: учебник. М., 2009.

7. *Гвоздева В. А.* Информатика. Часть I. Курс лекций. М., 2009.

8. *Гвоздева В. А.* Информатика. Часть II. Методические рекомендации по выполнению лабораторных работ. М., 2009.

9. *Безручко В. Т.* Практикум по курсу «Информатика»: учеб. пособие. М., 2002.

10. *Могилев А. В., Пак Н. И., Хеннер Е. К.* Информатика: учеб. пособие. М., 2000.

11. *Назаров С. В., Мельников П. П.* Программирование на MS Visual Basic: учеб. пособие. М., 2003.

12. *Попов И. И., Партыка Т. Л.* Операционные системы, среды и оболочки. М., 2003.

13. *Симонович С. В., Евсеев Г. А.* и др. Информатика. Базовый курс: учебник для вузов. 2-е изд. / под ред. С. В. Симоновича. СПб., 2007.

14. *Симонович С. В., Евсеев Г. А., Алексеев А.* Специальная информатика: учеб. пособие. М., 2001.

15. *Туркин О. В.* VBA. Практическое программирование. М., 2007.

16. *Угринович Н., Босова Л., Михайлова Н.* Практикум по информатике и информационным технологиям: учеб. пособие. М., 2001.

17. *Фигурнов В. Э.* IBM PC для пользователя. Краткий курс. М., 2001.

18. *Гайдамакин Н. А.* Автоматизированные информационные системы, базы и банки данных: учеб. пособие. М., 2002.

19. *Бажин И. И.* Информационные системы менеджмента. М., 2000.

20. *Крылова Г. Д.* Основы стандартизации, сертификации и метрологии: учебник. М., 2000.

21. *Дик В. В.* Информационные системы в экономике. М., 1996.

22. *Сафонов В. О.* Экспертные системы — интеллектуальные помощники специалистов. СПб., 1992.

23. *Агальцов В. П.* Базы данных. М., 2002.

24. *Избачков Ю. С.* Информационные системы: учебник. СПб., 2005.

25. *Когаловский М. В.* Перспективные технологии информационных систем. М., 2003.

Сайты Интернета:

1. www.google.ru
2. http://www.archeo.ru/rus/projects/archeograf.htm и др.

Оглавление

Введение 3

Часть I
ИНФОРМАТИКА

Глава 1. ОСНОВНЫЕ ПОНЯТИЯ ИНФОРМАТИКИ 9

1.1. История развития информатики 9
1.2. Основные понятия и определения 20
1.3. Информация и информационные процессы 21
1.4. Кодирование данных 23
1.5. Единицы представления, измерения и хранения информации 27
1.5.1. Вероятностный подход измерения количества информации 28
1.5.2. Объемный подход измерения количества информации 30
1.6. Системы счисления 31
1.6.1. Перевод чисел из одной системы счисления в другую 33
1.7. Логические основы ЭВМ 36

Глава 2. ОСНОВЫ РАБОТЫ С ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКОЙ 47

2.1. Основные этапы развития вычислительной техники 47
2.2. Архитектуры ЭВМ 54
2.3. Принципы работы вычислительной системы 57
2.4. Состав и назначение основных элементов персонального компьютера 60
2.5. Устройства ввода/вывода данных 68
2.6. Запоминающие устройства 71

2.7. Понятие системного и служебного (сервисного) программного обеспечения. Операционные системы 75
2.8. Файловая структура ОС. Операции с файлами 79
2.9. ОС MS-DOS и Windows. Работа с файлами и каталогами 86
2.10. Установка и обновление программного обеспечения 93
2.11. Работа с программами-архиваторами. Антивирусные программы 96

Глава 3. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ 107

3.1. Алгоритмизация 107
3.1.1. Виды алгоритмов и способы их записи 108
3.2. Технология программирования 114
3.3. Языки программирования 131
3.4. Система программирования Visual Basic 152
3.5. Язык программирования Visual Basic 163
3.6. Объекты, свойства, методы и события MS Visual Basic 178
3.7. Программирование в среде MS Visual Basic 187
3.7.1. Разработка первого проекта 187
3.7.2. Создание проектов 193
3.7.3. Программирование на рабочем листе Excel и в документе MS Word 200
3.7.4. Создание меню в MS Visual Basic 211
3.7.5. Работа с массивами в среде Visual Basic 216
3.7.6. Создание проектов с графикой, анимацией и звуком в MS Visual Basic 6.0 221

Часть II
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Глава 4. ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ 235

4.1. Информация и информационные технологии. Сферы применения 235

4.2. Этапы развития информационных технологий. Информационные революции 240
4.3. Информационная культура 242

Глава 5. ТЕХНОЛОГИИ ОБРАБОТКИ ИНФОРМАЦИИ . . . 244

5.1. Технологии обработки текстовой информации 244
5.1.1. Ввод и редактирование большого текста в MS Word, стилевое форматирование и оформление. Создание сложного документа 254
5.1.2. Создание составного документа (главного и вложенных). Макетирование страниц. Оформление титульного листа 259
5.2. Электронные таблицы Excel. Работа с данными и расчеты в Excel 263
5.2.1. Заполнение таблиц в Excel, редактирование, форматирование, фильтрация элементов и данных 273
5.2.2. Создание списков. Расчеты с использованием логических операций в Excel 277
5.2.3. Решение задач в Excel с использованием инструментов Подбор параметра и Поиск решения 282
5.2.4. Построение диаграмм в Excel. Интеграция приложений 286
5.3. Технологии обработки графической информации. Технология мультимедиа 289
5.3.1. Создание сложного рисунка в векторном редакторе Word и растровом редакторе Paint 300
5.3.2. Рисование сложного рисунка в растровом редакторе Photoshop 306
5.3.3. Система MS PowerPoint 311

Глава 6. ОСНОВЫ БАЗ ДАННЫХ И ЗНАНИЙ. СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ (СУБД) 316

6.1. Создание БД под управлением СУБД MS Access 328

6.2. Создание и использование запросов в БД 335
6.3. Вычисления в запросах. Создание отчетов 342

Глава 7. СЕТИ ЭВМ, ВЕБ-ТЕХНОЛОГИИ 348
7.1. Локальные и глобальные сети ЭВМ 348
7.1.1. Работа в Интернете. Службы Интернета 360
7.2. Создание веб-документов 369
7.3. Создание сообщений для электронной почты 371

Часть III
АВТОМАТИЗИРОВАННЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ (АИС)

Глава 8. ОБЩАЯ ХАРАКТЕРИСТИКА АИС 377
8.1. Основные сведения об АИС, понятия и определения 377
8.2. Состав и структура АИС 385
8.3. Методы, стадии и этапы создания АИС 394

Глава 9. ОБЕСПЕЧИВАЮЩАЯ И ФУНКЦИОНАЛЬНАЯ ЧАСТИ АИС 401
9.1. Информационное обеспечение 401
9.2. Математическое обеспечение 420
9.3. Программное обеспечение 441
9.4. Техническое обеспечение 450
9.5. Правовое, организационное, методическое и эргономическое обеспечение 463
9.6. Функциональные подсистемы АИС 470
9.6.1. Примеры функционирующих АИС 480

Глава 10. ТИПЫ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ И ПЕРСПЕКТИВЫ ИХ РАЗВИТИЯ 490
10.1. Типы АИС 490
10.1.1. Система «КонсультантПлюс» 517
10.2. Направления развития АИС 523

Заключение 533

Литература 536

Учебное издание

Гвоздева Валентина Александровна

Информатика, автоматизированные информационные технологии и системы

Редактор *Е. А. Тульсанова*
Корректор *О. Н. Картамышева*
Компьютерная верстка *И. В. Кондратьевой*
Оформление серии *Т. В. Иваншиной*

Подписано в печать 28.10.2014. Формат 60×90/16.
Гарнитура «Таймс». Усл. печ. л. 34,0. Уч.-изд. л. 34,6.
Бумага офсетная. Тираж 400 экз. Заказ № 6168

ООО «Издательский Дом «ФОРУМ»
101990, Москва — Центр, Колпачный пер., д. 9а
Тел./факс: (495) 625-39-27
E-mail: forum-books@mail.ru

ООО «Научно-издательский центр ИНФРА-М»
127282, Москва, Полярная ул., д. 31в, стр.1
Тел.: (495) 280-15-96, 280-33-86
Факс: (495) 280-36-29
E-mail: books@infra-m.ru
Http://www.infra-m.ru

По вопросам приобретения книг обращайтесь:

Отдел продаж «ИНФРА-М»
127282, Москва, ул. Полярная, д. 31в
Тел.: (495) 280-15-96
Факс: (495) 280-36-29
E-mail: books@infra-m.ru

Отпечатано способом ролевой струйной печати
в ОАО «Первая Образцовая типография»
Филиал «Чеховский Печатный Двор»
142300, Московская область, г. Чехов, ул. Полиграфистов, д.1
Сайт: www.chpd.ru, E-mail: sales@chpd.ru, т/ф. 8(496)726-54-10

КНИГИ

Издательского Дома «ФОРУМ»

право
•
экономика
•
психология
•
педагогика
•
техническая
литература
•
информационные
и компьютерные
технологии

УЧЕБНИКИ
для вузов,
техникумов,
колледжей и лицеев
ПОСОБИЯ
для поступающих
в вузы
СПРАВОЧНАЯ
ЛИТЕРАТУРА

Приглашаем к сотрудничеству

ПО ВОПРОСАМ ПРИОБРЕТЕНИЯ
ЛИТЕРАТУРЫ И
С ПРЕДЛОЖЕНИЯМИ
ПО ИЗДАНИЯМ
ПРОСИМ ОБРАЩАТЬСЯ
ПО АДРЕСУ:
101990, Г. МОСКВА-ЦЕНТР,
КОЛПАЧНЫЙ ПЕР., 9А.
ТЕЛ.: (495) 625-39-27,
ФАКС: (495) 625-39-27,
E-MAIL: FORUM-BOOKS@MAIL.RU

ИЗДАТЕЛЬСКИЙ ДОМ «ФОРУМ»

сообщает о выходе в серии
«Профессиональное образование»
следующих книг:

Агальцов В. П., Титов В. М. Информатика для экономистов. 338 с.

Безручко В. Т. Информатика. Курс лекций. 432 с.

Безручко В. Т. Компьютерный практикум по курсу «Информатика». 386 с. + CD.

Гвоздева В. А., Лаврентьева И. Ю. Основы построения автоматизированных информационных систем. 320 с.

Гришин В. Н. Панфилова Е. Е. Информационные технологии в профессиональной деятельности. 416 с.

Канцедал С. А. Алгоритмизация и программирование. 352 с.

Колдаев В. Д. Основы алгоритмизации и программирования. 416 с.

Колдаев В. Д., Лукин С. А. Архитектура ЭВМ. 384 с.

Колдаев В. Д., Павлова Е. Ю. Сборник задач и упражнений по информатике. 256 с.

Макаров А. А., Тюрин Ю. Н. Анализ данных на компьютере. 368 с.

Немцова Т. И., Голова С. Ю., Абрамова И. В. Программирование на языке высокого уровня. Программирование на языке Object Pascal. 496 с.

Основы компьютерных сетей / под ред. Л. Г. Гагариной. 272 с.

Технология разработки программного обеспечения / под ред. Л. Г. Гагариной. 400 с.

Федотова Е. Л. Информационные технологии в профессиональной деятельности. 368 с.

Черников Б. В. Информационные технологии управления. 352 с.

Шангин В. Ф. Информационная безопасность компьютерных систем и сетей. 416 с.

По вопросам приобретения литературы и с предложениями по изданиям просим обращаться по адресу:

101990, г. Москва-Центр, Колпачный пер., 9а.
Тел./факс: (495) 625-39-27
e-mail: forum-books@mail.ru

УВ 9504 сч.