Профессиональное образование

М. С. Цветкова
Л. С. Великович

Информатика и ИКТ

Учебник

Общеобразовательные дисциплины



ACADEMIA

ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ

М. С. ЦВЕТКОВА, Л. С. ВЕЛИКОВИЧ

Информатика и ИКТ

УЧЕБНИК

Рекомендовано
Федеральным государственным учреждением
«Федеральный институт развития образования»
в качестве учебника для использования в учебном процессе
образовательных учреждений, реализующих программы
среднего (полного) общего образования в пределах основных
профессиональных образовательных программ НПО и СПО
с учетом профиля профессионального образования

Регистрационный номер рецензии 350
от 4 октября 2010 г. ФГУ «ФИРО»

6-е издание, стереотипное

ACADEMIA

Москва
Издательский центр «Академия»
2014

УДК 621.391 (075.32)
ББК 32.81я723
Ц274

Рецензент —
зав. лабораторией Центра компьютерного обучения
Московского автомобилестроительного колледжа
при Академии народного хозяйства
при Правительстве Российской Федерации *А. А. Соломашкин*

Цветкова М. С.

Ц274 Информатика и ИКТ : учебник для сред. проф. образования / М. С. Цветкова, Л. С. Великович. — 6-е изд., стер. — М. : Издательский центр «Академия», 2014. — 352 с., [8] л. цв. ил.

ISBN 978-5-4468-0860-1

Дано понятие информационных процессов. Рассмотрены информационные модели, системы счисления, технологии программирования, принципы кодирования, хранения, поиска и обработки информации. Приведены основы алгоритмизации, даны примеры алгоритмов обработки информации. Описаны средства информационных и телекоммуникационных технологий. Подробно изложена технология создания и преобразования информационных объектов (обработка текста, графического и табличного материала, звуковой информации, создание мультимедийной презентации).

Для студентов учреждений среднего профессионального образования.

УДК 621.391 (075.32)
ББК 32.81я723

Условные обозначения:

? — вопросы и задания;

П — проектное задание для всех профилей;

П-СЭ — проектное задание для социально-экономического профиля;

П-Т — проектное задание для технического профиля;

П-ЕН — проектное задание для естественно-научного профиля;

П-Г — проектное задание для гуманитарного профиля

Оригинал-макет данного издания является собственностью Издательского центра «Академия», и его воспроизведение любым способом без согласия правообладателя запрещается

© Цветкова М. С., Великович Л. С., 2011
© Образовательно-издательский центр «Академия», 2011
© Оформление. Издательский центр «Академия», 2011

SBN 978-5-4468-0860-1

ПРЕДИСЛОВИЕ

В последние годы происходят значительные изменения в содержании и характере труда квалифицированных рабочих и техников, связанные в первую очередь с усилением процессов информатизации и автоматизации производственных технологий, требующих информационной подготовки.

Учебник разработан с учетом возможности изучения курса информатики как на базовом, так и на профильном уровне. Он включает в себя следующие главы: информационная деятельность человека; информация и информационные процессы; средства информационных и коммуникационных технологий; технологии создания и преобразования информационных объектов; технологии работы с информационными структурами — электронными таблицами и базами данных, телекоммуникационные технологии. К каждой главе учебника предложены вопросы и задания, которые требуют самостоятельной проработки учащимися. Важную практическую роль играют проектные задания, которые направлены на применение полученных умений в реальной образовательной деятельности обучаемых, что позволяет гибко настраивать обучение на соответствующее профильное направление. Материал для углубленного изучения и соответствующие задания профильной направленности реализуют деятельностные аспекты обучения, помогают учащимся на практике применить ИКТ в будущей профессии. В соответствии с профилем обучения образовательные учреждения вправе выбрать тематику и количество проектных заданий. Сайты поддержки лицензионного программного обеспечения Microsoft и его технического сопровождения, а также сайты свободно распространяемого программного обеспечения и ссылки на триальные версии различных информационных систем приведены в приложении.

Учебник предназначен для изучения курса «Информатика и ИКТ» как общеобразовательной дисциплины в учреждениях начального и среднего профессионального образования в соответствии с примерной программой курса.

ГЛАВА 1 ИНФОРМАЦИОННАЯ ДЕЯТЕЛЬНОСТЬ ЧЕЛОВЕКА

1.1. Основные этапы информационного развития общества

Современное общество часто называют информационным, и это совсем не случайно. В наше время информация стала базовым ресурсом общества наряду с традиционными — трудом, землей и капиталом. Однако ошибочным было бы полагать, что информационное общество стало зарождаться, как это принято считать, с середины XX в. Можно смело утверждать, что информация стала жизненно необходимым ресурсом с самого появления разумной жизни на земле. Самые древние люди использовали «информационные системы», которые предупреждали их об опасности, помогали им заготавливать пищу, обеспечивали общение между ними и т.д.

При раскопках первобытных стоянок археологи иногда находили камни, которые явно не использовались в качестве орудий труда или строительного материала. Как правило, они были примерно одинакового размера и располагались кучками. Было высказано предположение, что эти камни применялись для учета чего-либо, например туш животных или необходимого количества шкур. Может быть, это и было первое искусственное «запоминающее устройство».

С высокой долей достоверности можно предположить, что систем счета тогда еще не было, но понятие «столько-сколько» могло быть, поскольку использование простейших моделей свойственно человеку от природы, достаточно вспомнить игры маленьких детей, которые фактически являются не чем иным, как моделированием реальной жизни.

В истории человеческого общества несколько раз происходили существенные «прорывы», которые можно назвать *информационными*.

Первый информационный «прорыв» связан с *изобретением письменности*, что позволило накапливать и распространять знания. Ци-

вилизации, освоившие письменность, развивались быстрее других, достигали более высокого культурного и экономического уровня. Примерами могут служить Древний Египет, страны Междуречья, Китай. Позднее переход к алфавитному способу письма сделал письменность более доступной и способствовал смещению центров цивилизации в Европу (Греция, Рим).

За десятки тысяч лет до нашей эры стали появляться устройства, обеспечивающие «автоматизацию» счета, а за сотни лет — и первые счетные устройства — счеты. Такие находки были сделаны на территории Китая, Греции, Японии.

Второй информационный «прорыв» (в середине XVI в.) был связан с *изобретением книгопечатания*. Стало возможным не только сохранять информацию, но и обеспечить массовую доступность к ней, что, несомненно, повлияло на развитие науки и техники, помогло промышленной революции. Книги обеспечили интеллектуальную интеграцию различных культур, это способствовало ускорению создания общечеловеческой цивилизации. Фактически появилось первое средство *массовой коммуникации*.

Третьим информационным «прорывом» (в конце XIX в.) считается стремительный прогресс *средств связи*. Телеграф, телефон, радио позволили оперативно передавать информацию на любые расстояния, получать ее, выражаясь нынешним языком, в «реальном масштабе времени». Очевидно, что эти изобретения обусловил очередной этап технологической революции, связанный с открытием электричества. «Прорывы» в технологиях всегда предшествуют информационным «прорывам», создавая для них соответствующую материальную базу.

6 января 1838 г. была проведена первая успешная демонстрация передачи информации в кодированном виде. Последовательность букв, цифр и других символов в виде тире и точек получила название «азбука Морзе» в честь автора Сэмюэля Морзе.

14 февраля 1876 г. Александр Грэхем Белл (1847 – 1922) — профессор физиологии органов речи Бостонского университета, запатентовал в США свое изобретение — телефон.

Первая телефонная линия в Европе соединила 5 ноября 1877 г. берлинский почтамт с телеграфом (длина линии около 2 км). Первая телефонная станция была установлена в 1879 г. в Нью-Хевене. Одним из наиболее важных изобретений для дальнейшего усовершенствования телефонов было создание в 1878 г. английским физиком-изобретателем и профессором музыки Дейвидом Юзом угольного микрофона, в котором он в свою очередь использовал идеи американского изобретателя Томаса Эдисона (1847 – 1931) и российского ученого М. М. Михальского. Номеронабиратель был разработан в 1889 г. американцем Алмоном Строуджером, а первый телефон-автомат продемонстрирован на следующий год на Всемирной выставке в Париже.

Четверый «информационный прорыв» обусловлен в первую очередь созданием *электронных вычислительных машин* (ЭВМ), работающих в цифровом режиме (середина 1940-х гг.), а также *микропроцессорной техники* и созданных на ее базе *персональных компьютеров* (ПК). (В дальнейшем мы подробнее остановимся на основных этапах развития ЭВМ.) Вскоре после этого возникли как локальные (внутри учреждения, отрасли, учебного заведения), так и глобальные (Интернет) компьютерные сети, принципиально изменившие системы хранения и поиска информации.

Всемирная паутина — World Wide Web (WWW) — была разработана в 1992 г. Точно известен автор этой технологии представления информации в глобальной компьютерной сети — Тим Бернерс-Ли из Европейского центра ядерных исследований (*CERN*), расположенного в Женеве (Швейцария). Web-технология обеспечила лавинообразный рост Интернета (от англ. *Inter Net* — глобальная сеть) и тот океан информации, который мы видим сейчас. Обратимся к истории Интернета в СССР и России. Уже к осени 1990 г. сложилось ядро Сети СССР. В сентябре 1993 г. был зарегистрирован домен .SU, а 7 апреля 1994 г. — домен .RU, официально открывший существование Интернета в России. А в начале ноября того же года начал выходить первый в Рунете гипертекстовый журнал, т.е. заработал http-протокол, позволяющий обмениваться информацией в Интернете через сайты.

Глобальная сеть Интернет претендует на то, чтобы объединить все средства вещания и коммуникации, компьютерные, телефонные, радио- и видеосети, связав их в единое *киберпространство*.

Естественно, что информационное развитие общества носило поступательный характер, однако приведенные информационные «прорывы», обусловленные, с одной стороны, созревшими потребностями общества, а с другой — новыми технологическими возможностями, наиболее ярко характеризуют этот процесс. Накопленный к сегодняшнему дню колоссальный информационный потенциал и появление новых информационных и коммуникационных технологий не могли не сказаться на социально-экономическом характере современного общества. Если до середины XX в. общество имело ярко выраженный *индустриальный характер*, то нынешнее его состояние ученые характеризуют как *постиндустриальное*, рассматривая его в качестве переходного к *информационному обществу*.

Вопросы и задания

1. Какую роль играла и играет информация в развитии общества? Приведите примеры.
2. Почему появление письменности дало толчок развитию науки и культуры?

3. Как связаны развитие технологий и информационное развитие общества?
4. Почему создание книгопечатания можно рассматривать как этап массовых коммуникаций в развитии информационного общества?
5. Какие новые информационные возможности открыли перед обществом средства связи? Как это повлияло на экономическое развитие общества?

Проектное задание

Подберите материал для выполнения проекта «Плакат-схема. История развития информационного общества». Иллюстрации подберите в Интернете.

1.2. Роль информационной деятельности в современном обществе

Информационный «бум» кардинальным образом изменил современное общество, в том числе структуру экономики многих стран. Если раньше подавляющую часть экономики составляли промышленные предприятия — заводы и фабрики, а общество в целом так и называлось — *индустриальным*, то в наше время за счет широкого проникновения новых информационных технологий практически во все области жизни общества значительно изменилась не только экономическая, но и социальная структура общества, которое принято называть *постиндустриальным*. И действительно, в период индустриальной экономики инвестиционная стоимость предприятия в основном определялась стоимостью его материальных активов: станков, зданий, технологического оборудования и т.д. Сейчас информация наряду с трудом, землей и капиталом стала основной экономической категорией. Информационные технологии обусловили создание новой отрасли человеческой деятельности — в результате появились многочисленные IT-компании. Такие технологии активно внедряются в профессиональную деятельность, в значительной степени преобразуя характер труда. Естественно, что успехи информационных технологий достигнуты в первую очередь благодаря бурному развитию компьютерной техники. Компьютер уже прочно вошел и в жизнь человека. Способность компьютера хранить, передавать и обрабатывать колоссальные объемы информации обусловила кардинальное изменение характера информационной деятельности человека в самых разных областях.

С помощью компьютеров и соответствующего программного и информационного обеспечения ведется бухгалтерия предприятий, осуществляется документооборот, планирование, учет, контроль и анализ,

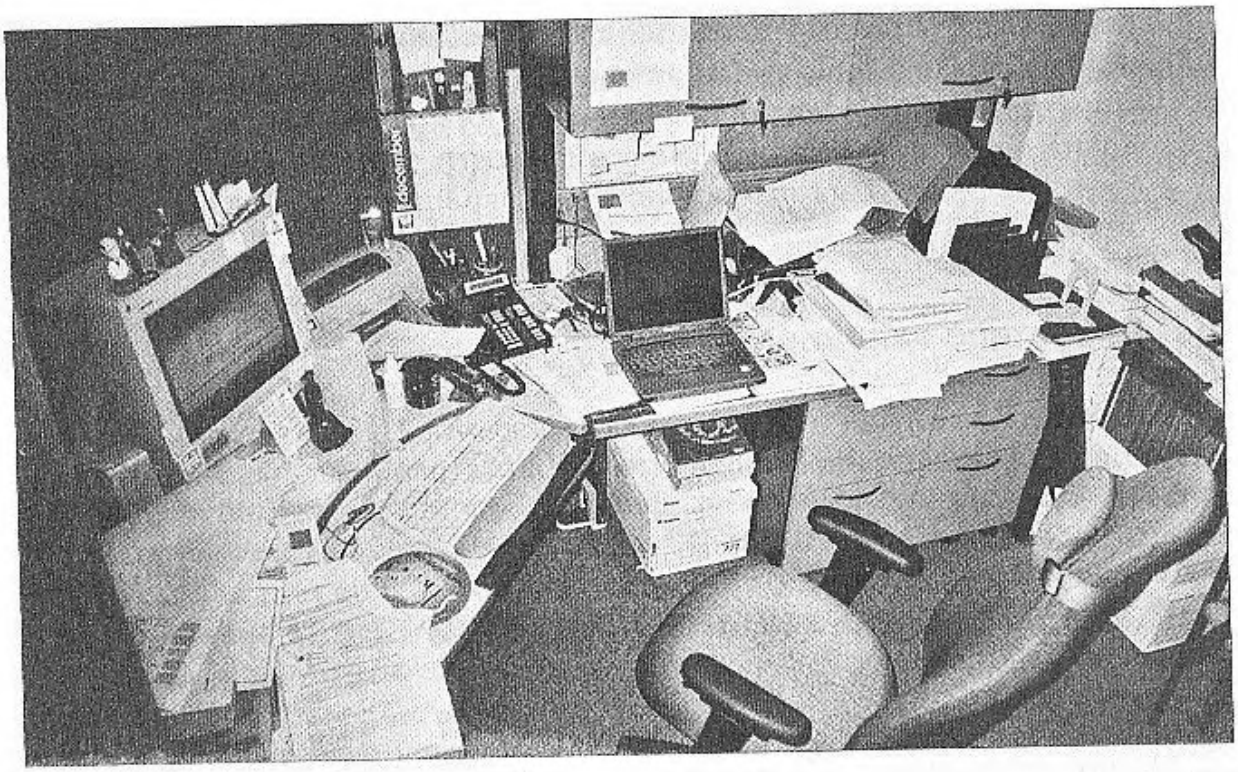


Рис. 1.1. Современный офис

обеспечивается электронная почта и связь с электронными массивами данных. Сети компьютеров связывают разных пользователей, расположенных в одном учреждении или находящихся в различных регионах страны и мира (рис. 1.1). Компьютеры находят применение при выполнении широкого круга производственных задач. Например, диспетчер на крупном заводе имеет в своем распоряжении автоматизированную систему контроля, обеспечивающую бесперебойную работу различных агрегатов (рис. 1.2).

Диспетчер гидро-, тепловой или атомной электростанции с помощью компьютерной системы контроля видит состояние всей технологической цепочки. Пилот самолета контролирует состояние оборудования и параметры полета с помощью автоматизированной информационной системы. Даже современный автомобиль немыслим без компьютерного блока. Компьютеры управляют роботами и манипуляторами. **Робот** — это механическое устройство, управляемое компьютером, например на линиях сборки автомобилей (рис. 1.3). Роботы выполняют многократно повторяющиеся операции, в частности затягивание болтов или окраску деталей кузова. Компьютеры ни на мгновение не теряют внимания к производственному процессу и не нуждаются в перерывах на обед.

Роботы могут также выполнять работу, которая оказывается для людей слишком тяжелой или вообще невозможной, например в условиях сильной жары или лютого мороза. Они готовят опасные химические препараты, работают в сильно загрязненном воздухе, в полнейшей

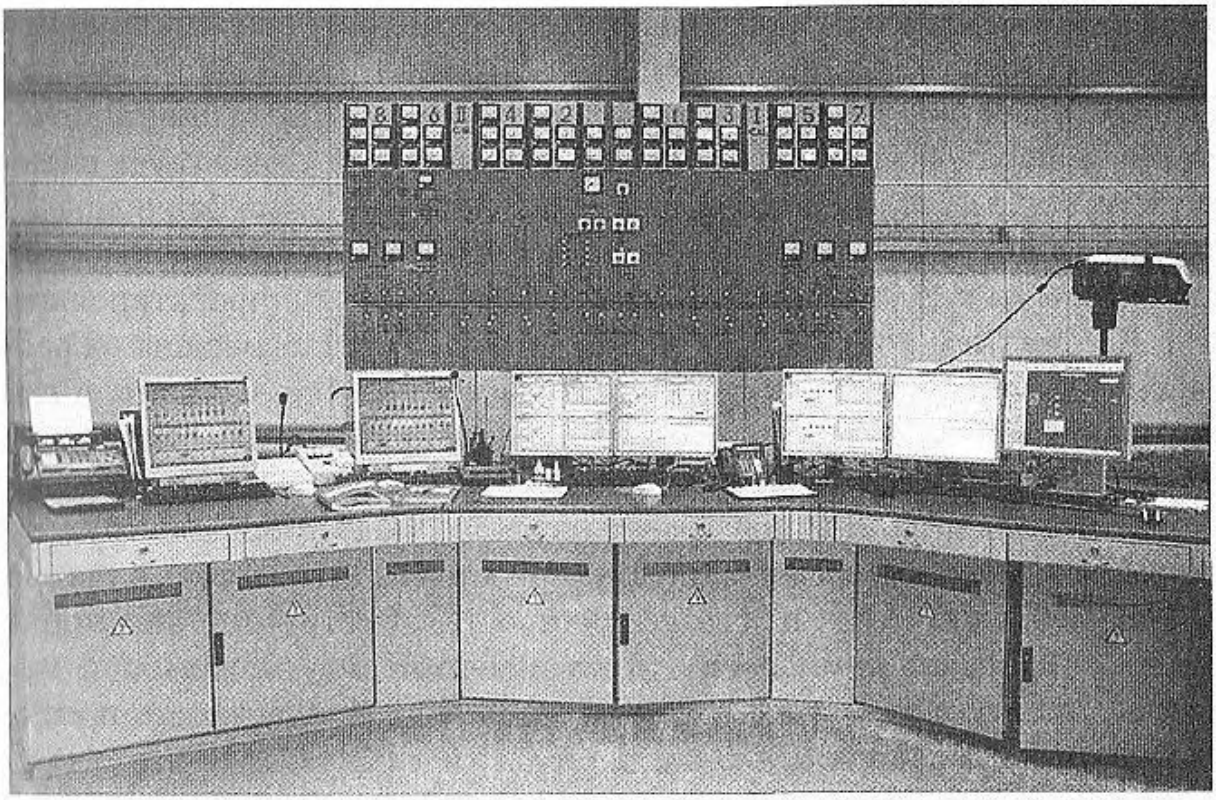


Рис. 1.2. Пульт управления электростанцией

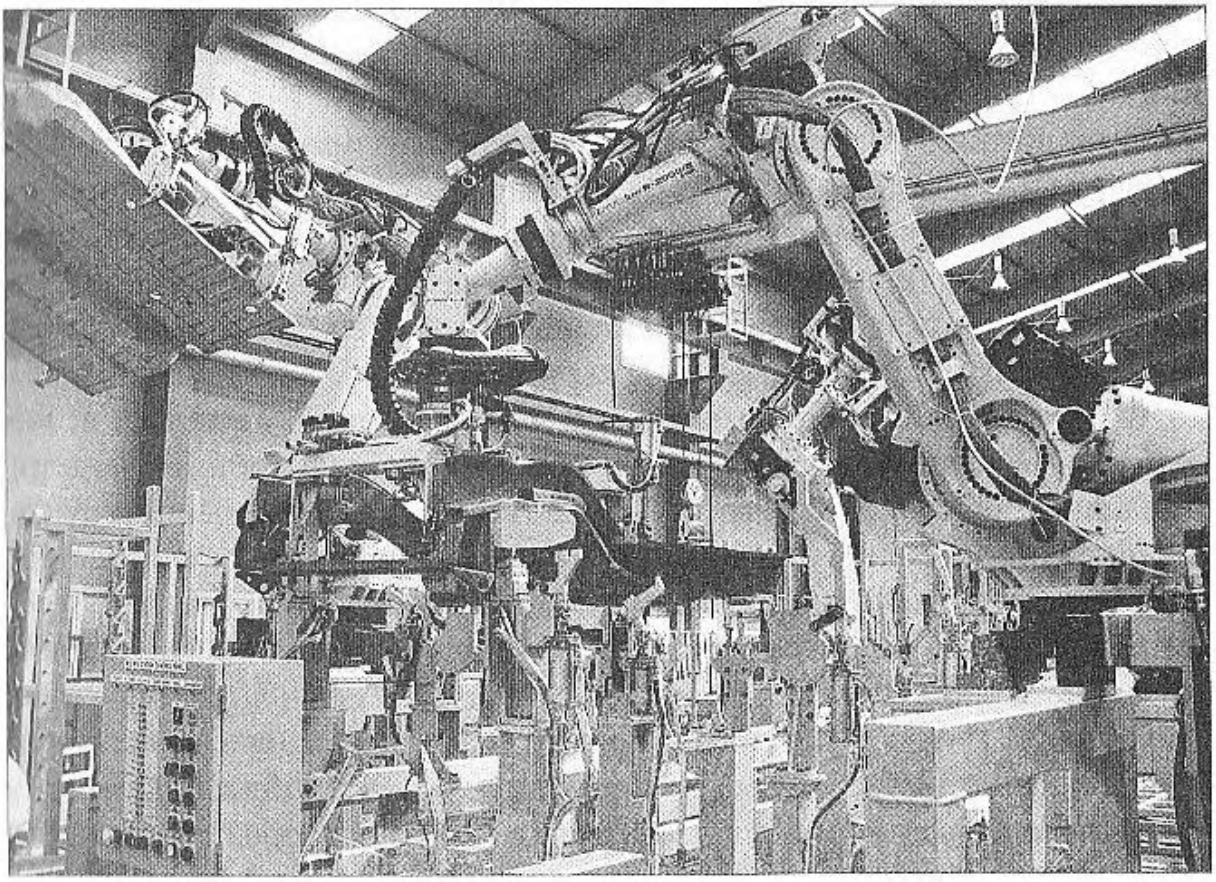


Рис. 1.3. Промышленный робот

темноте, во взрывоопасных зонах. Нередко один робот может заменить на заводе нескольких рабочих.

Как известно, созданию новой продукции, будь то машина, механизм, аппарат, а также здания или сооружения, предшествует очень ответственный этап проектирования. Конструкторские бюро еще 20 лет назад выглядели как длинные ряды кульманов (чертежных досок), за которыми в белых халатах стояли конструкторы и проектировщики. Коллективы конструкторов и инженеров тратили месяцы на расчеты, изготовление чертежей для сложных проектов. Сегодня, в век компьютеров, конструкторы имеют возможность целиком посвятить свое время процессу конструирования, поскольку расчеты и подготовку чертежей компьютер «берет на себя». Например, конструктор корабля может непосредственно за компьютером исследовать влияние формы обводов судна на его устойчивость, конструктор автомобиля —производить компоновку узлов и механизмов создаваемой машины, архитектор — исследовать прочность и долговечность проектируемого здания, и т. д.

В основе проектирования и исследования характеристик создаваемого объекта центральную роль занимает *математическое моделирование*, при котором исследуются математические зависимости, отражающие практически все параметры и характеристики проектируемого объекта. Компьютерные системы, обеспечивающие такой процесс, называются **системами автоматизированного проектирования** (САПР).

Признанная во всем мире система проектирования AutoCAD (http://usa.autodesk.com/autocad/) является одним из самых мощных и универсальных средств разработки проектной и конструкторской документации, создания эскизов и чертежей разнопрофильного назначения, схем, деталировок, трехмерных твердотельных и поверхностных моделей (рис. 1.4).

С помощью САПР повышается производительность труда, и, конечно, не нужно объяснять, насколько лучше электронный аналог бумажного чертежа. В AutoCAD имеются специализированные надстройки, предназначенные для решения задач в конкретной сфере деятельности (машиностроение, архитектура и строительство, электроника и др.). Популярным средством в нынешней архитектуре является семейство программных комплексов ArchiCAD (http://www.archicad.ru/company/buy.html). Эти пакеты охватывают практически весь процесс создания архитектурного проекта — от настройки параметров программы, проекта и инструментов до создания и редактирования основных и дополнительных конструктивных элементов здания на чертежах и в 3D-окне, а также использование библиотек и дополнений,

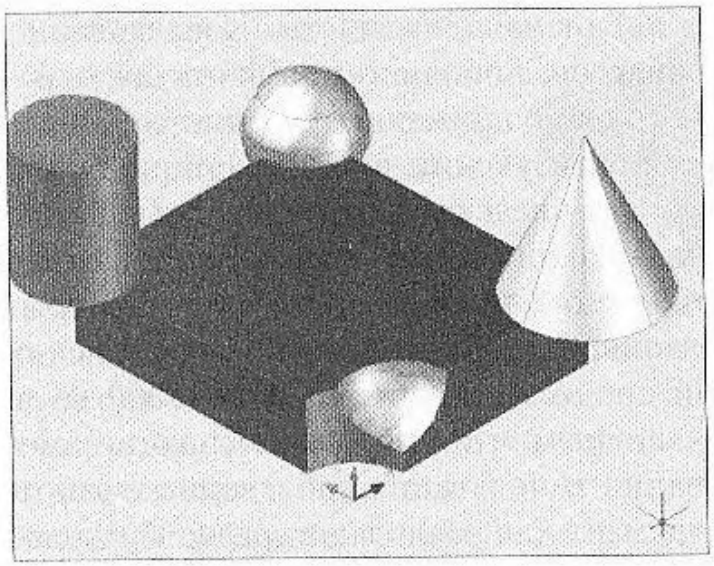


Рис. 1.4. Окно программы AutoCAD

изготовление смет и спецификаций, создание презентаций и вывод проекта на печать, способы коллективной работы.

На основе САПР конструкторы одновременно проектируют отдельные детали и элементы общего изделия, товара, конструкции и сооружения. Проектирование ведется с последующей компоновкой деталей в общую сборку.

Компьютерные системы используются для управления *строительным процессом*, доставкой блоков на строительные площадки, а также в целях учета оборудования, объемов выполненных работ, финансового учета.

Мощные компьютерные системы позволяют выполнять большое число *банковских операций*, включая обработку чеков, регистрацию изменения каждого вклада, прием и выдачу вкладов, оформление ссуды и перевод вкладов с одного счета на другой и (или) из банка в банк. Кроме того, крупнейшие банки имеют автоматические устройства, расположенные за пределами банка. Банковские автоматы (банкоматы) созданы для того, чтобы брать деньги со счета в любое время дня и ночи. Все, что требуется, — вставить пластмассовую банковскую карточку в автоматическое устройство и указать код. Как только это сделано, необходимые операции будут выполнены.

Современная наука также немыслима без информационных технологий. В *фундаментальной науке* компьютеры используют как для моделирования явлений и процессов, так и для обработки результатов экспериментов. Научные искусственные спутники Земли посредством оптических, локационных и других приборов собирают информацию о ближнем и дальнем космосе и передают ее на Землю, где с помощью компьютерных систем воссоздаются картины «солнечного ветра», метеоритной угрозы, процессов, происходящих на расстоянии многих

тысяч световых лет от нашей планеты. Исследование Земли, ее геофизических процессов, породило новую науку — *геоинформатику*, методы которой успешно применяются как при разведке новых месторождений, так и при изучении процессов, происходящих в глубине Земли, что позволяет предсказывать природные катаклизмы, обоснованно планировать экологические мероприятия. Сегодня нет ни одной научной области, где бы не применялись компьютеры.

Информатизация, затронувшая практически все области знаний, вовлекла в свой процесс на рубеже 1980—1990 гг. и *историческую науку*. Стало очевидным, что ПК — не только незаменимое средство для редактирования и печатания собственных работ историков, но и удобный инструмент для реализации разнообразных источниковедческих методик. Появление *сканеров* — считывающих устройств, создающих возможности для оптического ввода в память компьютера текста или изобразительного материала, — дало новый импульс исследованиям по истории культуры, искусства, в музейном и архивном деле. Было разработано специализированное программное обеспечение, учитывающее особенности исторических источников, что привело к активизации деятельности целого ряда научных центров по созданию больших баз данных (БД) (рис. 1.5).

Компьютеры давно вошли в *сферу торговли*. Когда покупатели выкладывают свои покупки на прилавок, кассир пропускает каждую из них через оптическое сканирующее устройство, которое считывает

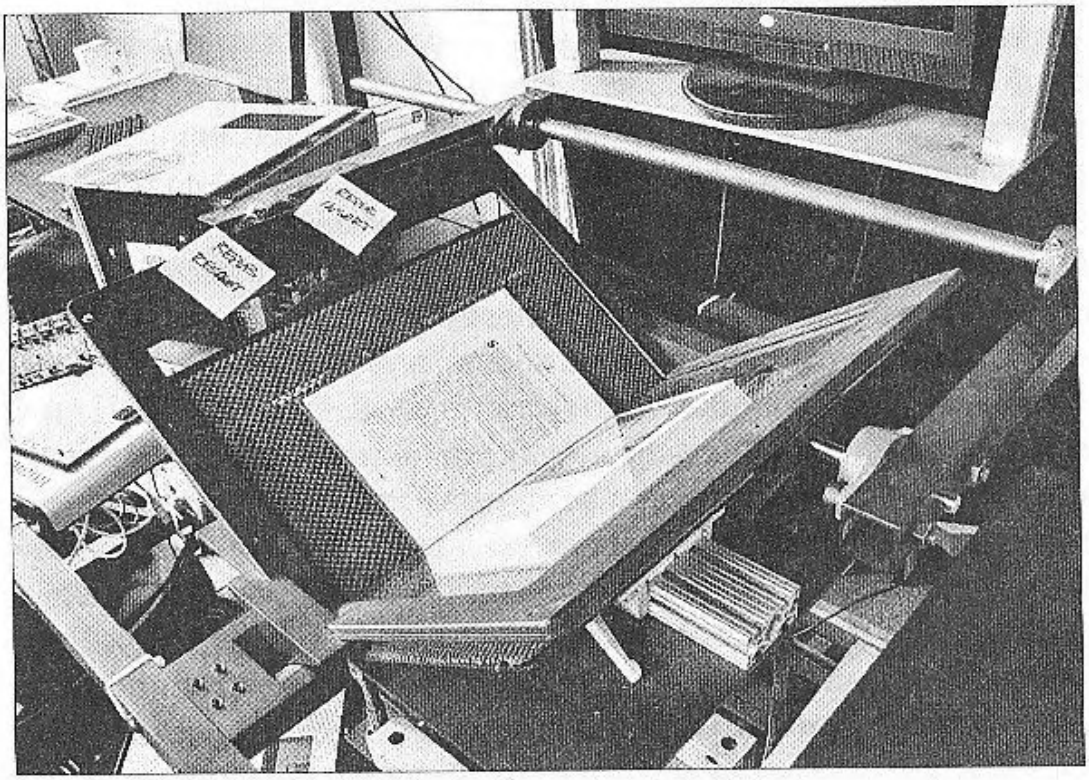


Рис. 1.5. Книжный сканер в библиотеке

уникальный код, нанесенный на покупку (рис. 1.6). *Уникальный код* — это серия точек и цифр, по которым компьютер определяет товар; цена этого товара хранится в памяти компьютера и высвечивается на экране кассового аппарата, чтобы покупатель мог видеть стоимость своей покупки. Как только все отобранные товары прошли через оптическое сканирующее устройство, компьютер немедленно выдает общую стоимость купленных товаров. В этом случае окончательный расчет с покупателями происходит намного быстрее, чем при использовании обычного кассового аппарата. Применение компьютера не только позволяет существенно ускорить расчет с покупателями, но и дает возможность все время держать под контролем количество проданного и имеющегося в наличии товара.



Рис. 1.6. Сканер штрих-кода для работы кассира

Современное *образование* тоже немыслимо без применения компьютеров. Компьютеры помогают моделировать явления и процессы, осваивать теоретический материал, осуществлять эффективный контроль уровня знаний учащихся. Используя сайт образовательного учреждения, можно записаться на лабораторные работы, посмотреть расписание, разместить ответ на вопрос на специальном форуме, записаться на курс дистанционного обучения и многое другое.

В чем же состоит причина такой популярности информационных технологий в образовании? Прежде всего в том, что компьютер позволяет в значительной мере индивидуализировать учебный процесс, выбрать темп обучения, степень подробности, дает возможность еще и еще раз вернуться к материалу, проверить уровень понимания его и т. д. Компьютерное моделирование дает возможность провести реально невыполнимые в обычных условиях эксперименты. Например, научиться анализировать кровообращение в организме человека, управлять государством, расщеплять атомное ядро или путешествовать по миру.

Компьютеры в учебных заведениях используются не только для обучения, но и для управления образовательным процессом. Администрация школы, например, на основе **автоматизированной системы управления** (АСУ) и БД учреждения составляет расписание занятий, отчеты по успеваемости, сводки по учащимся, учитывает материальные ценности, выполняет бухгалтерские расчеты. Автоматизация этой ру-

тинной работы значительно повышает скорость ее выполнения и снижает трудозатраты.

Во многих образовательных учреждениях созданы локальные компьютерные сети, имеющие выход в Интернет, разработаны web-сайты, ведутся электронные журналы успеваемости, БД физического и психологического развития учащихся.

Особая роль принадлежит системам дистанционного обучения. Информационные технологии и ресурсы дистанционного обучения в Интернете являются на сегодня самым мощным средством самообразования и саморазвития людей, со школьной скамьи приобщая к этому процессу, без которого в современном информационном обществе невозможен путь к успеху в профессиональной сфере. Быстро меняющиеся технологии влекут за собой смену и появление новых профессий, требующих непрерывного обучения работников. Дистанционное обучение является важнейшим средством для самореализации людей с ограниченными физическими возможностями.

Медицина — очень сложная наука. И здесь ей на помощь приходит компьютер. Существует множество болезней, каждая из которых имеет только ей присущие симптомы. Кроме того, есть десятки болезней с похожими симптомами. В подобных случаях врачу бывает трудно поставить точный диагноз. В настоящее время многие врачи используют компьютерную диагностику. Предположим, у человека произошел сердечный приступ, и он находится в блоке интенсивной терапии. Здесь он «подключен» к компьютеру, который следит за числом сер-

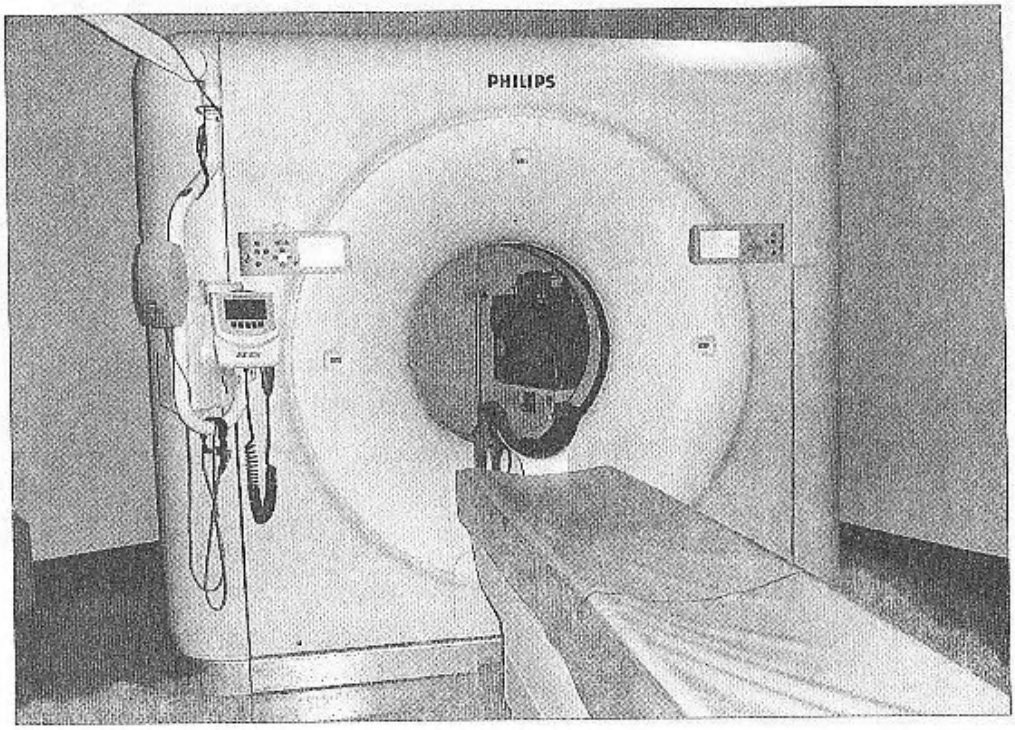


Рис. 1.7. Компьютерный томограф

дечных сокращений: если оно вдруг уменьшится до опасного уровня, компьютер немедленно сообщит об этом врачу или медицинской сестре. Конечно же, окончательное решение принимает врач, но компьютер ускоряет процесс принятия решения и может повлиять на точность диагностирования. Например, прибор, сканирующий внутренние органы человека, — томограф, дает точное изображение изменений внутренних органов человека (computer-aided tomography — компьютерная томография) (рис. 1.7). Компьютеры играют важную роль в медицинских исследованиях. Они позволяют установить, как влияет загрязнение воздуха на заболеваемость населения данного района. Кроме того, с их помощью можно изучать влияние нежелательных воздействий на человека, в частности последствия удара при автомобильной катастрофе для черепа и позвоночника человека.

Базы медицинских данных позволяют медикам быть в курсе последних научных и практических достижений. Компьютеры используются для создания карт, показывающих скорость распространения эпидемий. Компьютеры хранят в своей памяти истории болезни пациентов, что освобождает врачей от бумажной работы, на которую уходит много времени, и позволяет больше времени уделять больным (рис. 1.8).

Способность компьютеров хранить и быстро обрабатывать большое количество информации используется *правоохранительными органами.* Электронные хранилища больших массивов информации — банки данных доступны государственным и региональным следственным учреждениям всей страны.

Компьютеры применяются правоохранительными органами и в процессе розыскной работы. Например, в лабораториях криминалистов



Рис. 1.8. Электронная регистратура

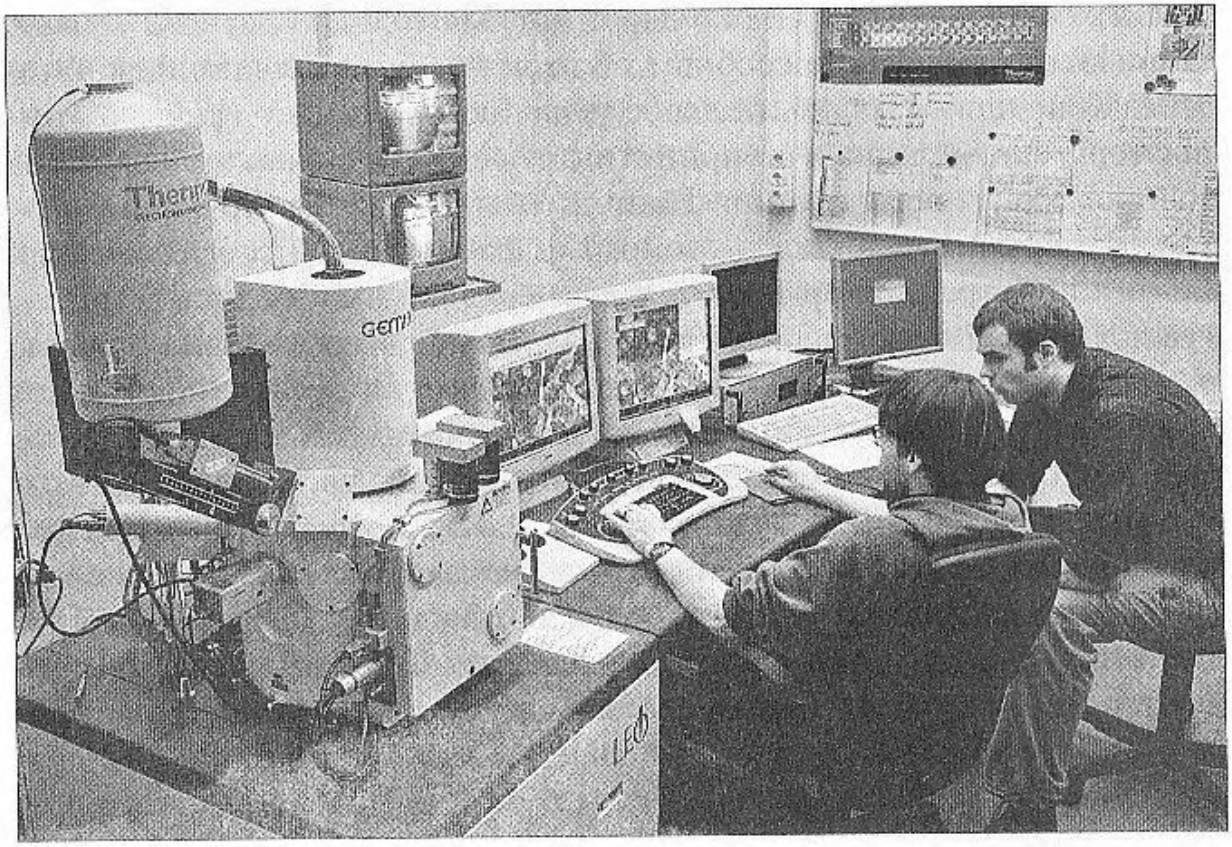


Рис. 1.9. Цифровой микроскоп

компьютеры помогают проводить анализ веществ, обнаруженных на месте преступления (рис. 1.9). Заключения компьютера-эксперта часто оказываются решающими в доказательствах по рассматриваемому делу.

На наших глазах происходит технологический «прорыв» *в сельском хозяйстве* — компьютеры и индивидуальные микродатчики позволяют контролировать состояние и режим каждого отдельного животного и растения. Это высвобождает значительные материальные и людские ресурсы, резко улучшает качество труда в сельском хозяйстве.

На некоторых фермах применяются сложные автоматизированные системы управления подачей корма скоту, температурным режимом и поливом растений в теплицах. На основе полученной от них информации фермер может сделать заключение о том, что какое-то животное заболело, поскольку оно лишилось аппетита и выданная ему порция корма осталась нетронутой. Компьютер помогает фермеру планировать свой бюджет и вести учет домашнего скота, а также имеющегося оборудования и запасов, финансовых средств (издержек и расходов), осуществлять планирование продаж, заготовок и поставок, проводить комплексный анализ эффективности финансово-хозяйственной деятельности своей фермы.

Современная армия немыслима без компьютерной техники. Каждый самолет, стратегическая ракета, корабль или подводная лодка, танк оснащены специализированным компьютерным оборудованием и информационными системами, которые позволяют эффективно ориентироваться на местности, осуществлять поиск и уничтожение различных объектов противника, оптимизировать различные параметры боевой работы.

Рассмотрим для примера крылатую ракету. Ее основное преимущество заключается не только в скорости, но и в том, что она может двигаться к цели на предельно малой высоте, что, естественно, затрудняет работу средств противовоздушной и противоракетной обороны по ее обнаружению и уничтожению. Но ведь на ее пути далеко не идеальный рельеф местности — встречаются и возвышенности, впадины и другие препятствия, и тем не менее крылатая ракета успешно их преодолевает. Дело в том, что ее бортовой компьютер заранее, до сантиметра, «знает» маршрут (рис. 1.10). Этот маршрут формируется на основании информации, полученной со спутников Земли. Именно он закладывается в память бортового компьютера.

Несколько десятилетий назад компьютерами пользовались только ученые и математики. Сегодня же информационные технологии стали достоянием писателей, художников, музыкантов и представителей других профессий *мира искусств*. Компьютер помогает писать книги,

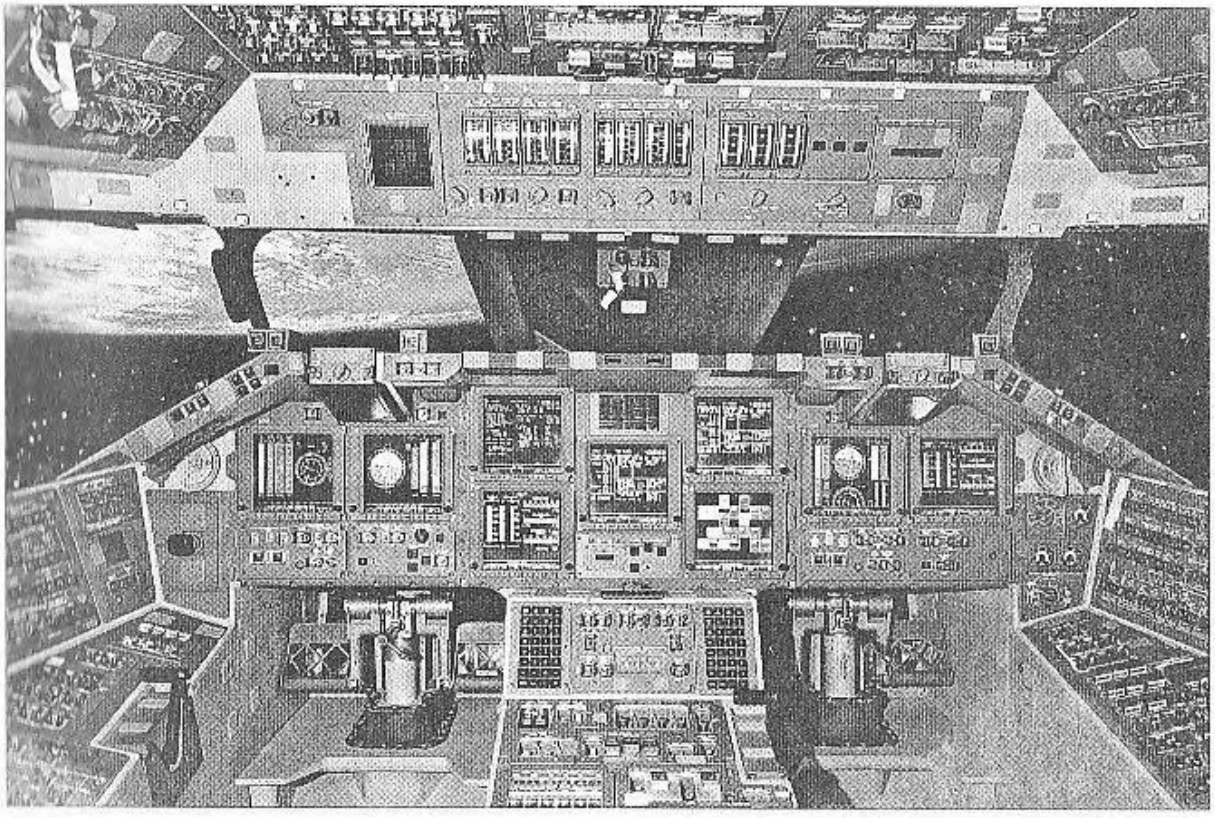


Рис. 1.10. Панель бортового компьютера самолета

рисовать, сочинять песни, создавать специальные эффекты в научно-фантастических фильмах. В последнее время все больше профессиональных писателей применяют текстовые процессоры для повышения качества и ускорения работы над публикациями. Но не только они, но и журналисты, сценаристы, а также многие другие используют компьютеры при работе с текстами. Текстовый процессор значительно облегчает редактирование и сверку текстов. Кроме того, он освобождает от необходимости перепечатки текстов и тем самым экономит время. Наконец, применение специальных программ помогает выявлять и устранять орфографические и синтаксические ошибки.

В руках художника компьютер становится инструментом для рисования. Иллюстраторы, дизайнеры, карикатуристы, кинематографисты считают, что информационные технологии и современное цифровое оборудование (цифровые фото- и видеокамеры, синтезаторы звука и пр.) предоставляют им новые возможности в их творческой деятельности. С помощью таких средств, как графопостроитель, графический планшет, световое перо, художники создают многоцветные рисунки. Музыканты посредством компьютера создают аранжировки музыкальных произведений, формируют звучание аккомпанемента. С помощью компьютеров во время представлений устраивают различные световые и лазерные шоу.

Значительные изменения в профессиональной деятельности людей, обусловленные применением информационных технологий, не могли не повлечь за собой и изменения *в социальной сфере*. Одним из наиболее важных моментов является помощь инвалидам (рис. 1.11). Они могут использовать компьютер для обучения, работы, общения, оформления заказов на продукты и даже для проведения видеоигр и соревнований. Например, люди, у которых парализованы руки, могут работать на компьютере с помощью ног, используя для этого ножной выключатель, напоминающий педаль электрической швейной машины. Инвалиды с парализованными руками и ногами могут использовать устройства, которые вставляются в рот или прикрепляются к голове. Говорящий компьютер, т. е. снабженный синтезатором речи, позволяет слепым людям выполнять операции, которые невозможны без помощи компьютера. Благодаря компьютеру утраченную способность видеть они компенсируют способностью слышать. Для тех,

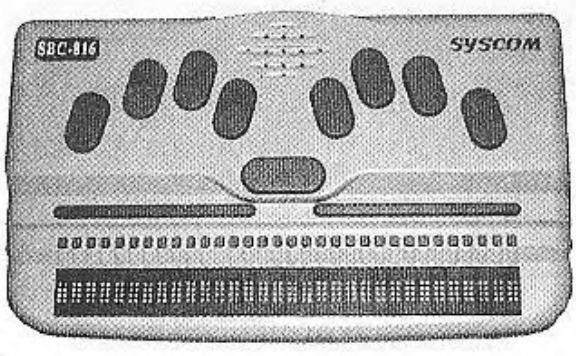


Рис. 1.11. Клавиатура для слепых

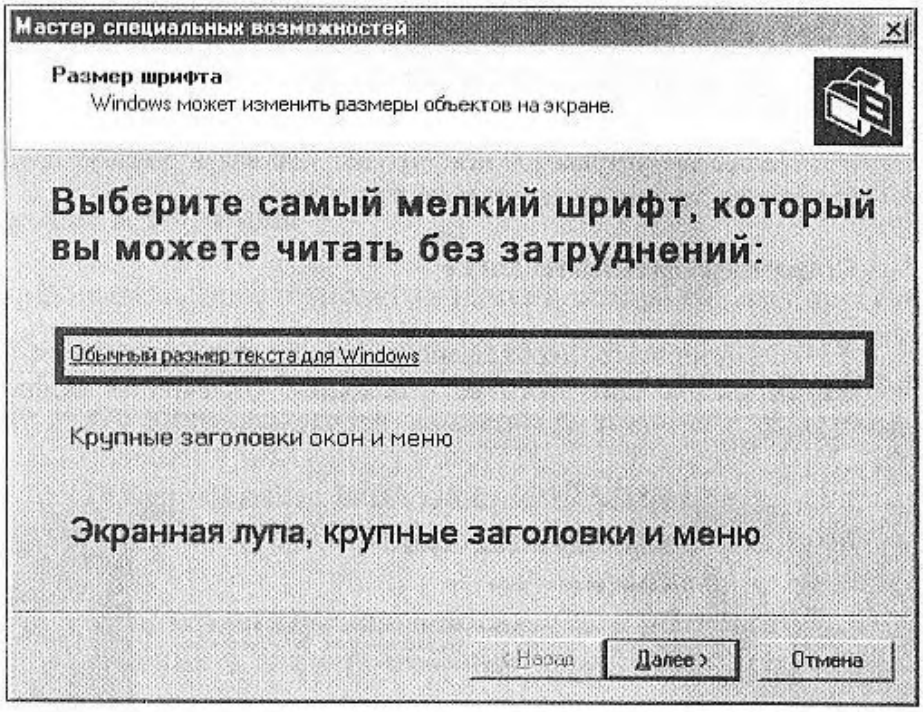


Рис. 1.12. Состав услуг мастера специальных возможностей

кто частично утратил зрение, имеется устройство, воспроизводящее текст в увеличенном масштабе, — экранная лупа (рис. 1.12, 1.13). Все это помогает инвалидам чувствовать себя полноценными членами общества, как это и должно быть в цивилизованном мире.

В последнее время компьютеры проникли в наш дом, причем в виде не только настольного персонального компьютера, но и «компьютера-невидимки». В ближайшее время подобные невидимки могут стать такими же существенными элементами нашей жизни, как и электричество. Что представляет собой компьютер-невидимка? Это крохотный микропроцессор, который «спрятан» в окружающих предметах. У большинства людей дома один-два таких компьютера, у некоторых — больше. Думаете, что это не так? Тогда ознакомьтесь с перечнем предметов, в каждом из которых может быть микропроцессор: электронные часы, телевизор, микроволновая печка, термостат, радиоприемник, телефон, калькулятор, посудомоечная машина, стиральная машина, швейная машина, мобильный телефон и т. д. И это только начало. Например, можно приспособить к входной двери звонок, который будет исполнять музыку по заданной программе.

Для каких же еще целей можно использовать дома компьютер? Сейчас, например, в некоторых домах компьютер применяют для поддержания нужной температуры и управления кондиционированием воздуха. Другие подключают к компьютерам устройства поливки приусадебных участков. Микрокомпьютеры применяются для включения и выключения электрического освещения в доме в соответствии

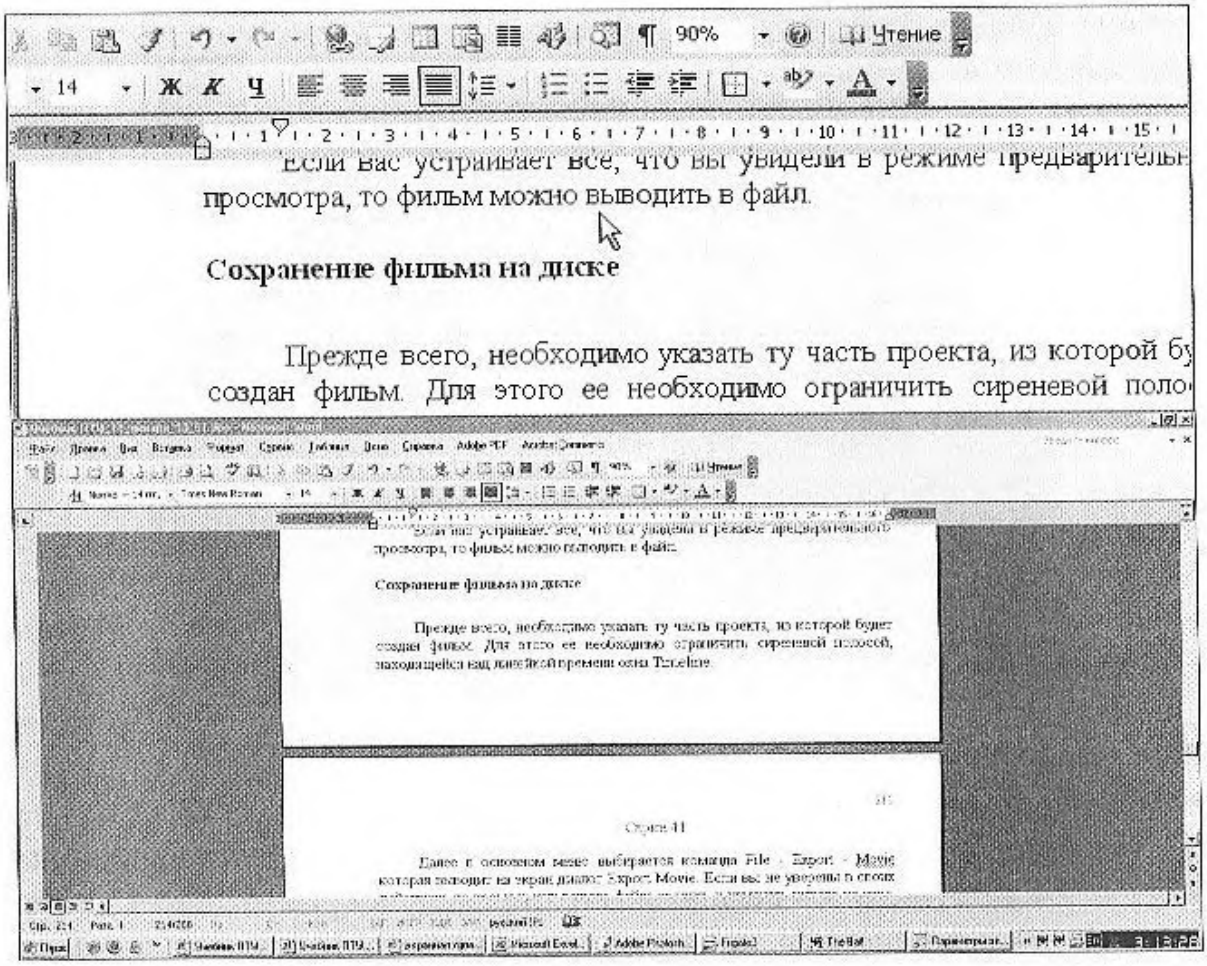


Рис. 1.13. Пример специальных возможностей «Экранная лупа»

с заданной программой, они управляют охранной сигнализацией и даже изменением цвета стен в интерьере и т. д. Существует концепция *электронного*, или «*умного*» дома. В таком доме все процессы, обеспечивающие комфортное существование людей, управляются компьютерными системами.

Информационные технологии коренным образом изменили не только быт людей и характер их профессиональной деятельности, но и психологию и образ жизни современных людей. Они разрушили границы общения, во много раз увеличили *коммуникативную активность людей*.

Глобальная компьютерная сеть и мобильная телефонная связь охватили весь мир. Интернет предоставляет уникальные возможности дешевой, надежной и конфиденциальной глобальной связи по всему миру как через компьютеры, так и через мобильные телефоны. Это оказывается очень удобным для фирм, имеющих свои филиалы по всему миру, сетевых сообществ и обычных людей. Человек, не выходя из дома, с помощью Интернета может работать, делать покупки, читать прессу, смотреть репортажи, учиться, заказывать и получать различные справки, проводить оплату налогов, коммунальных услуг, вести пере-

говоры. Интернет образует ядро, обеспечивающее связь различных информационных сетей, принадлежащих различным учреждениям и пользователям во всем мире.

? Вопросы и задания

1. Как информационные технологии изменили характер труда? В чем состоит автоматизация труда на современном производстве, в образовательных учреждениях, торговле, банках, медицине, а также в различных областях науки и искусства?
2. В каких областях применяют промышленных роботов, автоматизированные системы управления, компьютерную диагностику?
3. Какие функции конструктора (проектировщика) сегодня «взяли на себя» САПР?
4. Почему информационные технологии позволяют индивидуализировать процесс обучения и как они помогают реализовать потребности в самообразовании людей?
5. Как меняется уклад всей жизни человека современного общества? В чем заключается концепция электронного, или «умного», дома? Какова роль информационных технологий в росте коммуникативной активности людей?

П-СЭ Проектное задание

Подготовьте доклад «Новая экономика — экономика, основанная на информации и знаниях».

1.3. Информационные ресурсы общества

Информационные ресурсы общества представляют собой ресурсы, накопленные в форме, позволяющей их воспроизводство для общества, человека. Это ресурсы, несущие в себе информацию: книги, статьи, диссертации, научно-исследовательская и опытно-конструкторская документация, всевозможные БД и базы знаний, технические переводы, музейные коллекции, древние рукописи, археологические находки, произведения искусства, библиотечные фонды и архивы, кино-, фото-, аудиоматериалы, в том числе аудиокниги, и, конечно, материалы газет, журналов, телепередач, радиопрограмм и пр.

В отличие от всех других видов ресурсов (трудовых, энергетических, ископаемых и т. д.) информационные ресурсы тем быстрее растут, чем больше их расходуют. Отличительная черта и ценность основной массы информационных ресурсов — их *доступность* для простых пользователей. Каждый может стать пользователем этих ресурсов.

Возьмем самый обыденный пример. Допустим, врач выписал больному лекарство. Как мы знаем, нужное лекарство не всегда можно найти в ближайшей аптеке, да и «разброс» цен на одно и то же лекарство весьма существенен. Войдем на *поисковый сервер* в Интернете www.yandex.ru и сформируем *запрос*: «Наличие лекарств в аптеках Москвы».

В результате выполнения запроса появится перечень *сайтов* с интересующей нас информацией (рис. 1.14). Выберем один из них (рис. 1.15).

Войдя на выбранный сайт, в специальном *окне поиска* наберем название лекарства и при желании укажем ближайшую к вам улицу или станцию метро. В результате исполнения увидим перечень аптек, где есть это лекарство, его цену в каждой аптеке, а также узнаем, есть ли служба доставки на дом (рис. 1.16).

Конечно, к современным информационным ресурсам относят и те, которые хранятся в электронном виде. Человечество стремится сохранить свои информационные ресурсы в цифровом виде, тогда их можно использовать с помощью компьютеров. Все виды информационных ресурсов как в материальном, так и в электронном виде можно назвать «долговременной памятью человечества».

В множестве информационных ресурсов значительное место занимают *информационные образовательные ресурсы*. К ним относятся учебные книги, словари, энциклопедии, атласы, карты, различные

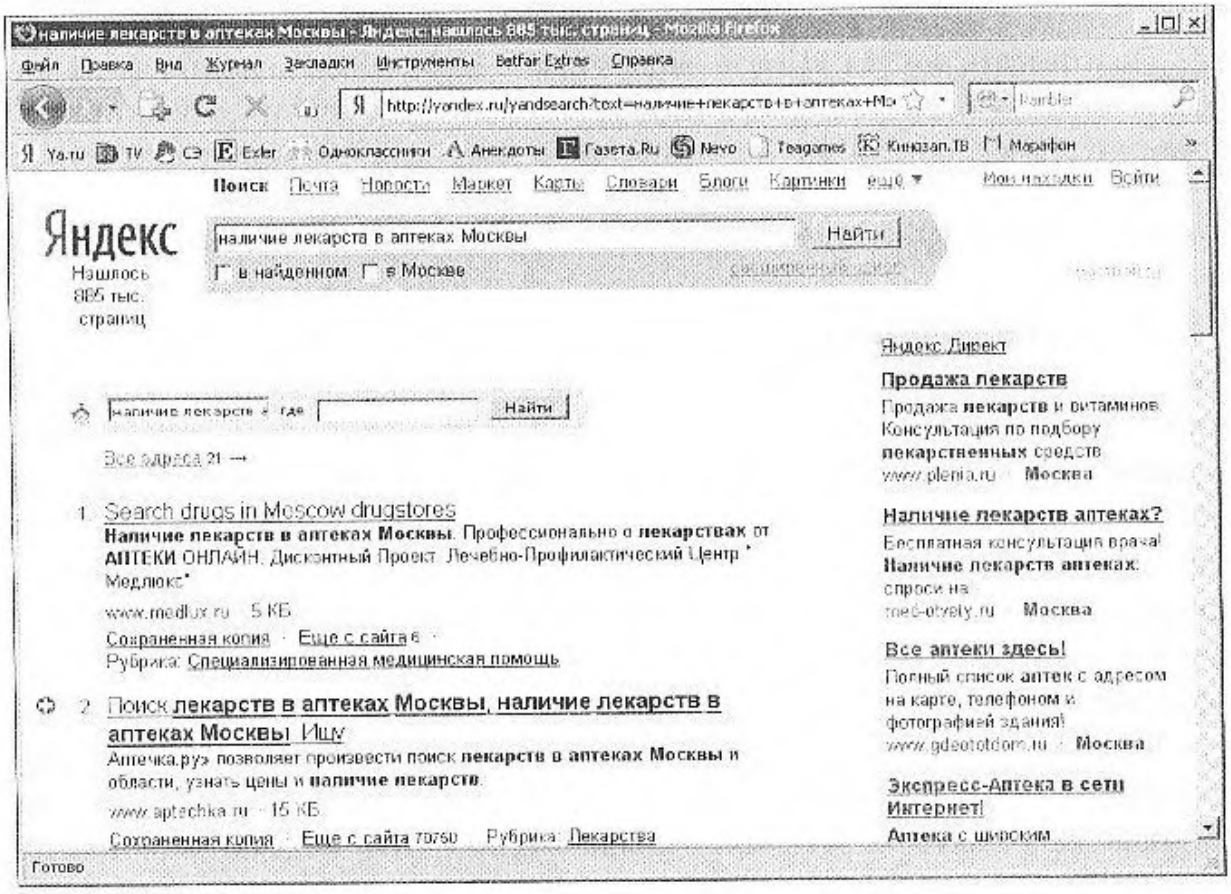


Рис. 1.14. Результат обработки запроса в поисковой системе «Яндекс»

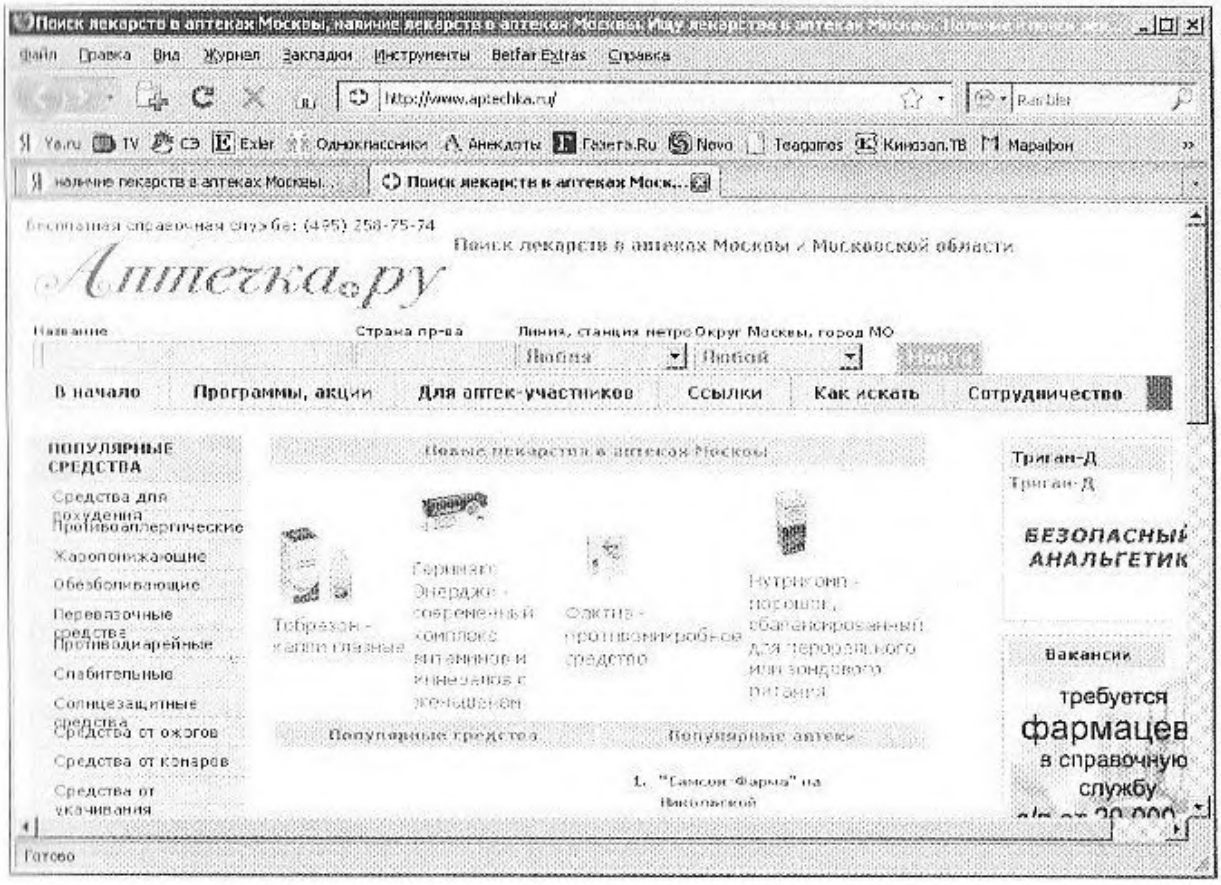


Рис. 1.15. Выбранный сайт

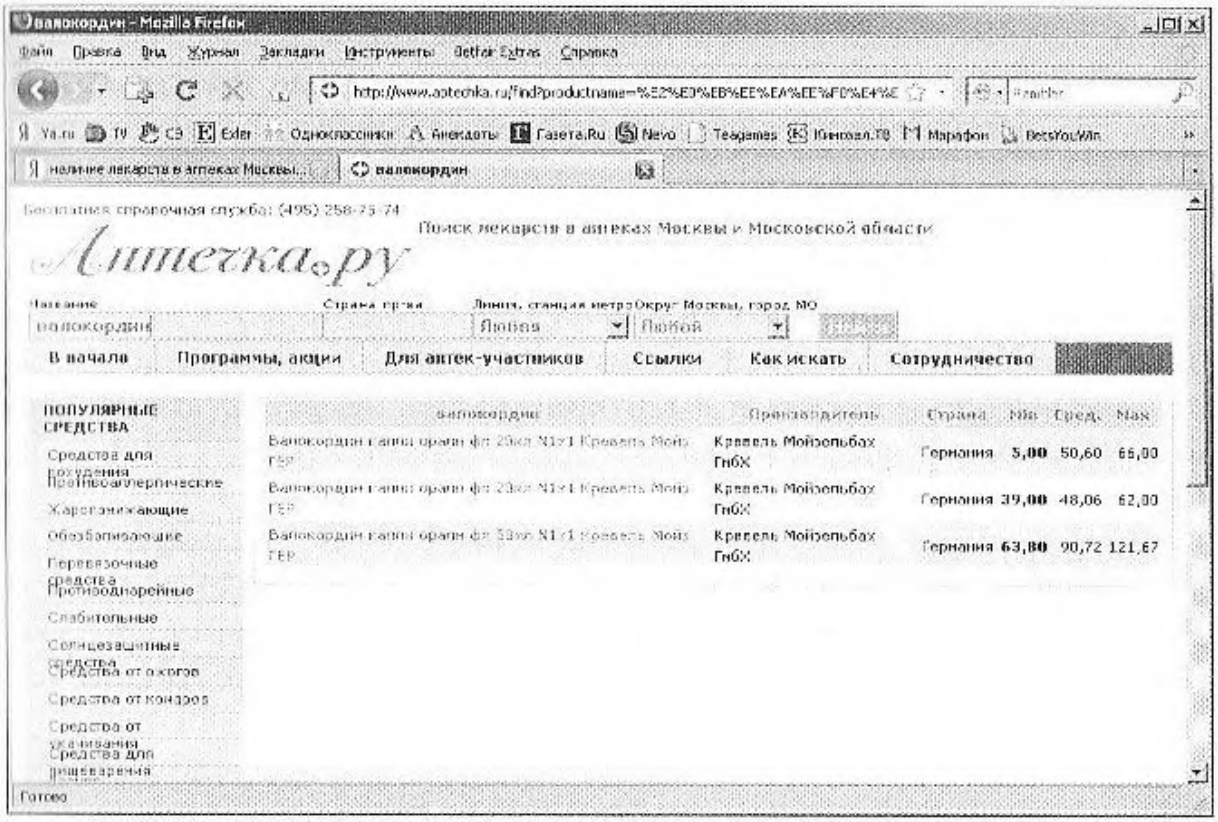


Рис. 1.16. Результат поиска лекарства



Рис. 1.17. Единое окно доступа к образовательным ресурсам

пособия и т. д. Однако в связи с прогрессом в области компьютерной техники появились новые формы представления учебной информации. В книжных магазинах помимо традиционной учебной продукции широко представлены компьютерные программные средства, используемые в образовательном процессе, разнообразные учебные материалы на компакт-дисках, содержащих обучающие, контролирующие и демонстрационные программы практически по всем образовательным разделам; электронные книги, словари-переводчики, электронные энциклопедии, объем которых в сотни раз превышает бумажную энциклопедию, различные юридические справочники (БД) на дисках и в Интернете. Эти же ресурсы в Интернете позволяют воспользоваться информацией из них в *онлайн-режиме*, т. е. режиме прямого доступа.

В странах мира разрабатываются образовательные коллекции общедоступных *цифровых образовательных ресурсов*. Для удобства доступа к ним в России организован *портал* (вход) на все образовательные сайты через *единое окно* http://window.edu.ru на государственном образовательном портале www.edu.ru (рис. 1.17).

Познакомиться с составом Единой коллекции цифровых образовательных ресурсов для школ нашей страны можно на сайте http://school-collection.edu.ru/.

Особое место в информационных ресурсах в последние годы стали занимать *электронные книги*, отличительной чертой которых является использование технологии *гиперссылок*, позволяющих переходить в установленных местах электронного текста (ссылках) на различные фрагменты книги напрямую.

Например, вы готовите курсовую или дипломную работу по литературе в текстовом редакторе. Ряд материалов у вас находится непосредственно на диске вашего компьютера. Для удобства в тексте курсовой работы можно сделать ссылку на цитируемую вами литературу или другой материал. Для этого можно выполнить команду текстового редактора **Вставка**, **Гиперссылка**. В открывшемся окне указать имя нужного файла, содержащего текст цитируемой вами книги или статьи, или имя файла с рисунком, схемой, аудиозаписью, видеороликом (рис. 1.18). Это имя файла в виде ссылки останется в тексте курсовой работы. Каждый раз, когда вам нужно будет обратиться к материалу из этого файла, следует при нажатой клавише **CTRL** щелкнуть левой клавишей мыши по ссылк,е и в вашем окне откроется нужный документ — текст, таблица, презентация, рисунок и т.д.

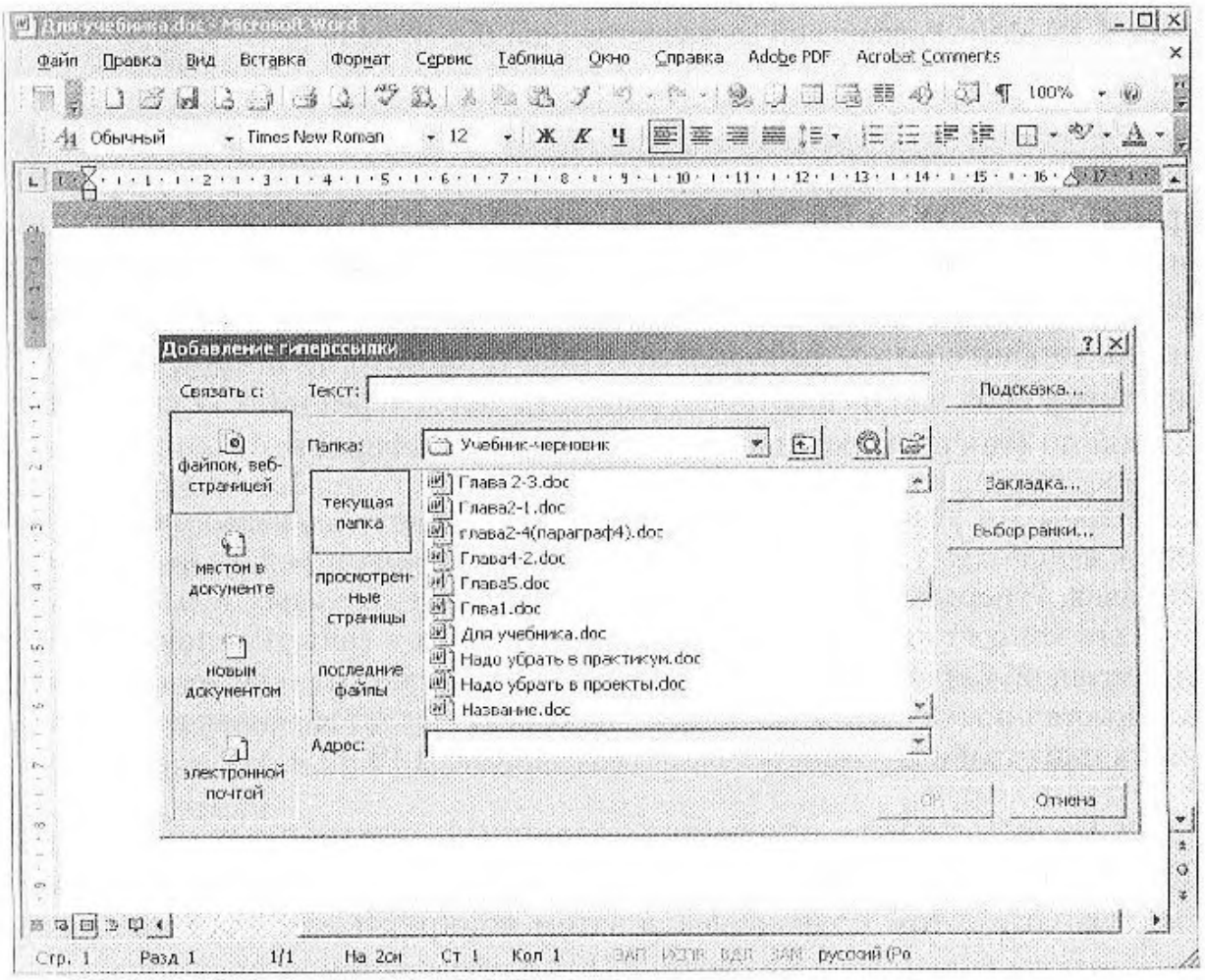


Рис. 1.18. Добавление гиперссылки в документ Word

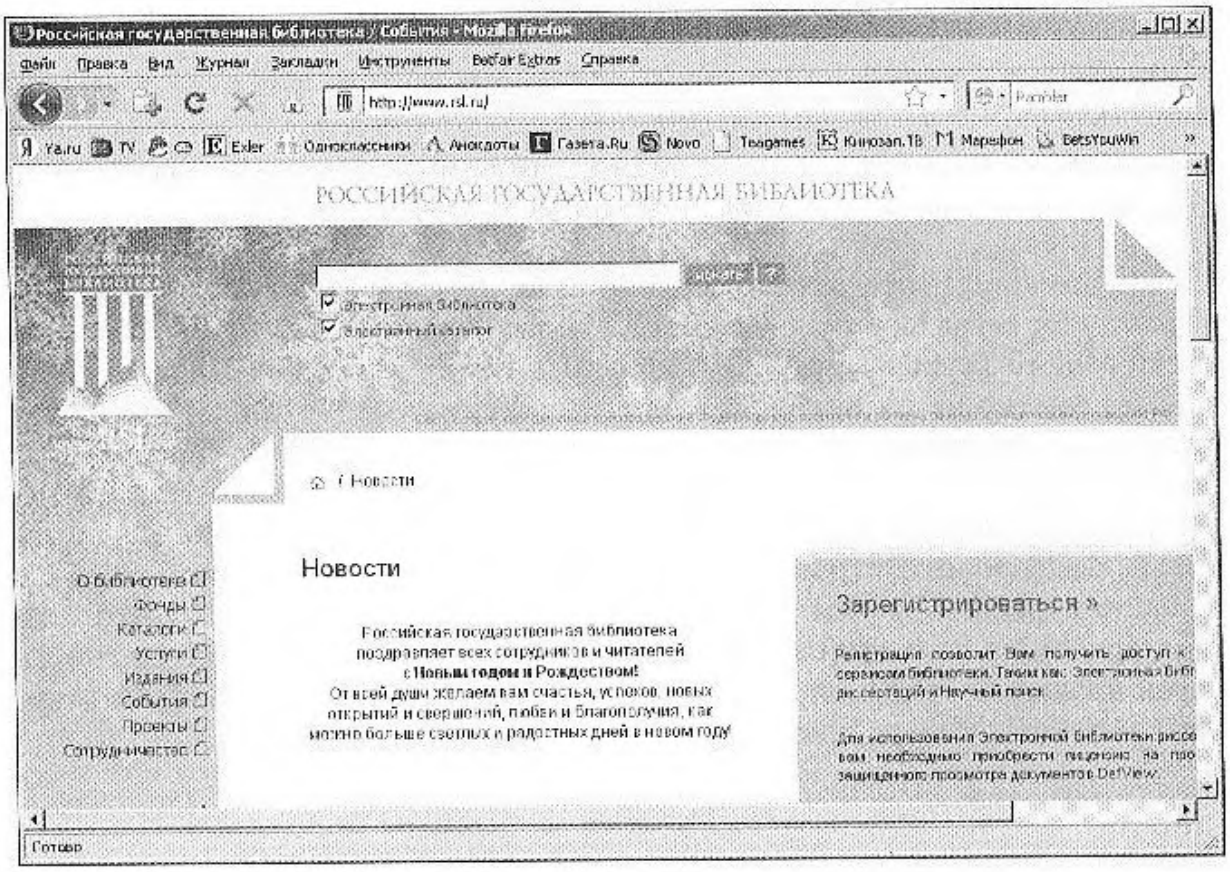


Рис. 1.19. Сайт Российской государственной библиотеки

Электронные библиотеки в системе информационных образовательных ресурсов стали неотъемлемой частью жизни — это не только файлы на диске компьютера или материалы на компакт-дисках, но и материалы на библиотечных сайтах в Интернете. Но каждый раз поиск таких материалов занимает достаточно много времени, поэтому очень удобно создать файл каталога электронной библиотеки по какой-то теме у себя на компьютере и работать далее через него, осуществляя вызов по гиперссылке какого-либо сайта в Интернете.

Чтобы это реализовать в вашей курсовой работе, сделаем в ней список ссылок используемых сайтов. Выполним в текстовом редакторе команду **Вставка**, **Гиперссылка** и в поле **Адрес** введем интернет-адрес нужного сайта или его конкретного раздела, например обратимся к сайту Российской государственной библиотеки http://www.rsl.ru. В результате, если, конечно, ваш компьютер подключен к Интернету, на экране появится головная страница сайта или страница заданного вами раздела (рис. 1.19—1.21).

В Интернете есть множество различных электронных библиотек. Каталоги электронных библиотек построены иерархически, т.е. сначала разделы, потом подразделы, а затем конкретные ссылки на статьи или тезисы (например, на сайте «ИКТ в образовании» http://www.ict.edu.ru/lib/) (рис. 1.22).

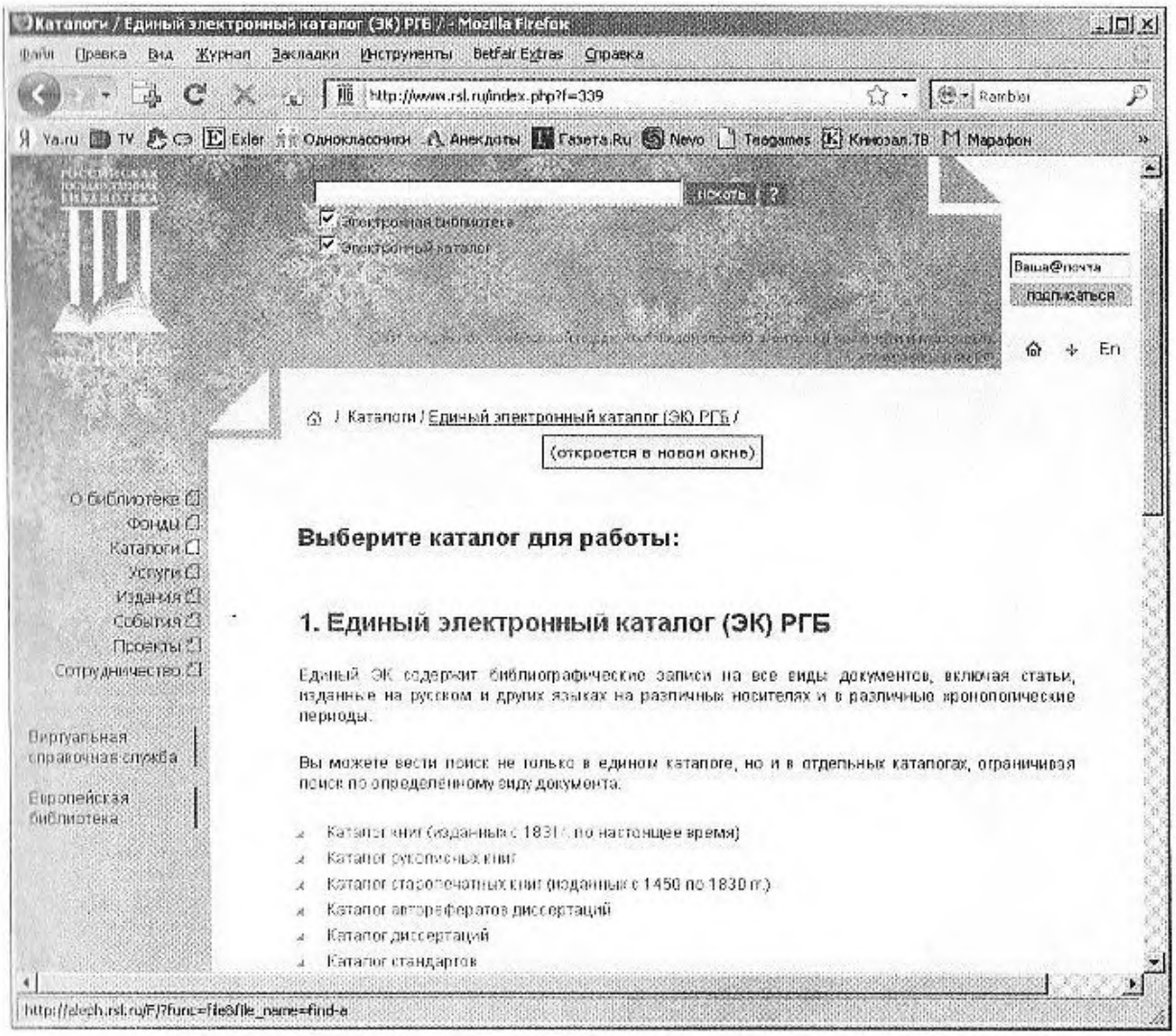


Рис. 1.20. Выбор каталога для работы

Вопросы и задания

1. Что относится к информационным ресурсам общества?
2. Составьте таблицу ссылок на сайты библиотек региона, в котором вы живете, используя Интернет.
3. Подберите коллекцию работ любимого художника, используя электронную экспозицию на сайте музея Третьяковской галереи (http://www.tretyakovgallery.ru/).
4. Осуществите перевод фраз в онлайн-режиме, используя сайт компьютерного переводчика Promt (http://www.promt.ru/).
5. Найдите закон об образовании, используя портал российского образования www.edu.ru.

Проектное задание

Используя сайт «Единое окно» (http://window.edu.ru), составьте список ссылок на ресурсы www.fcior.edu.ru (разделы НПО и СПО), имеющих непо-

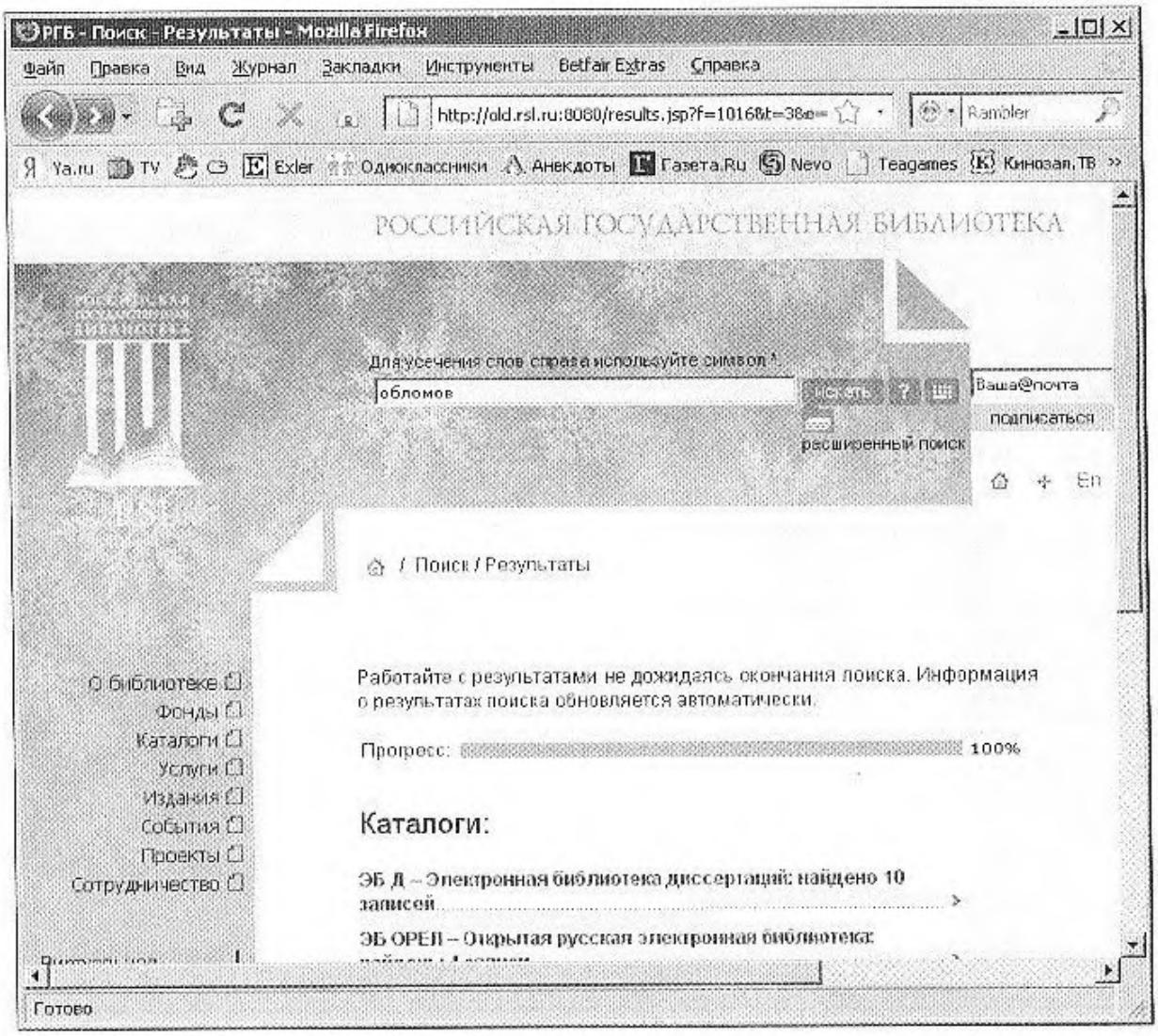


Рис. 1.21. Поиск книги в электронном каталоге

средственное отношение к подготовке по вашей будущей профессии. Воспользуйтесь также подборкой сайтов по профориентации (табл. 1.1).

Таблица 1.1

Описание ресурса	Имя сайта
Специальности системы профессионального образования (СПО). Этот раздел поможет вам узнать из действующих стандартов СПО о требованиях к выпускникам по выбранной специальности, о совокупности приобретенных в процессе обучения знаний, умений и навыков	http://www.edu.ru/abitur/act.11/index.php http://pedsovet.org/ секция ПТУ

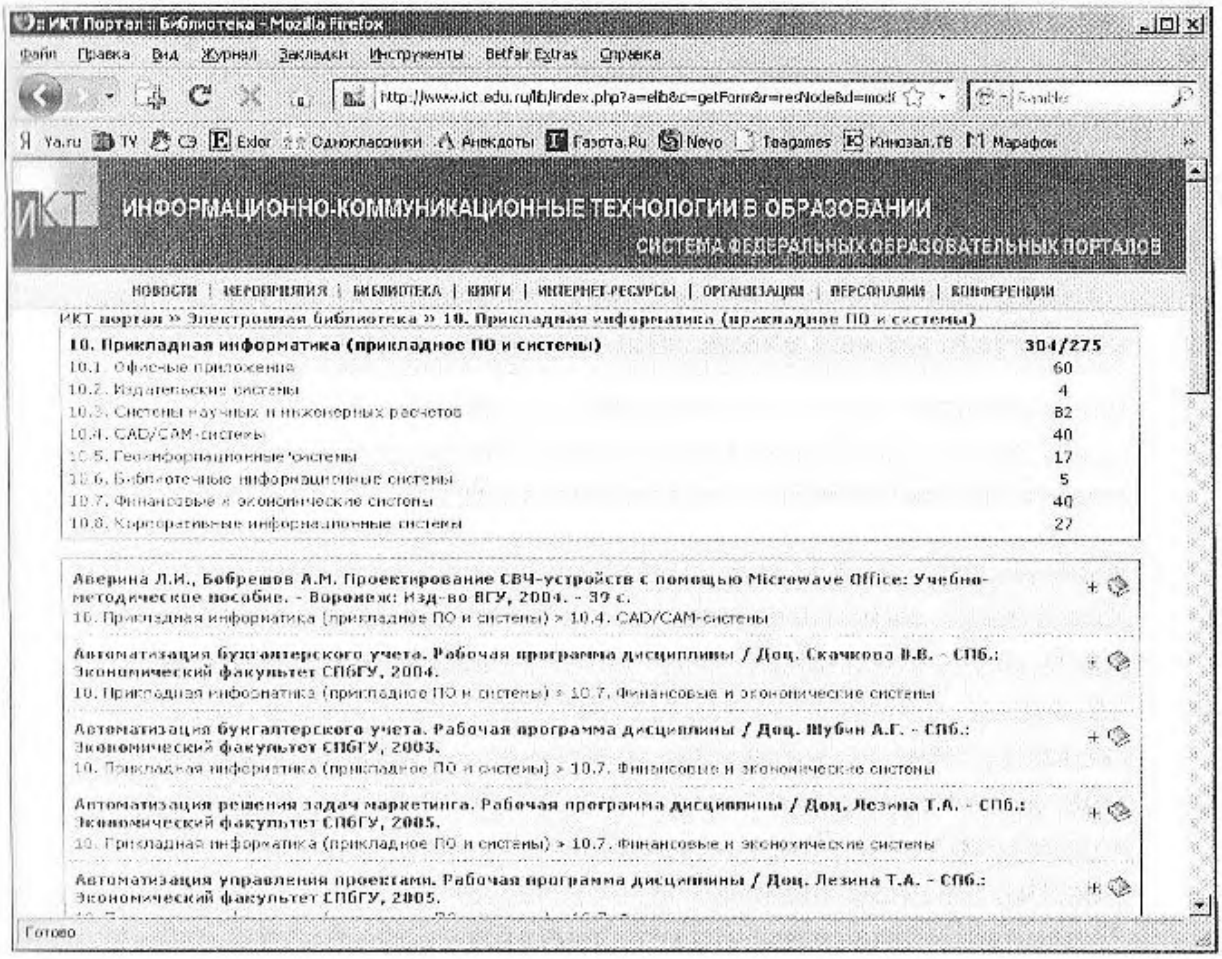


Рис. 1.22. Структура раздела электронной библиотеки ИКТ-портала

Окончание табл. 1.1

Описание ресурса	Имя сайта
Пси-фактор. Информационный ресурсный центр по практической психологии предлагает 20 тестов, часть из которых может оказаться полезной для лиц, выбирающих себе профессию	http://psyfactor.org/tests/
Справочник профессий. Приводится описание около 480 наименований профессий	http://www.rabotka.ru/infoworker/ http://www.rabota-in.net/prof/professions.php

ГЛАВА 2 ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

2.1. Информация и ее свойства

Информатика — это наука об информации, ее свойствах, видах, способах сбора, хранения, обработки, передачи, а также о соответствующем инструментарии. Мы часто слышим и используем термин «информация», но редко задумываемся, что же это такое на самом деле. В науке есть такое понятие, как *энтропия* — мера неопределенности. Фактически информация уменьшает энтропию, так как увеличивает объем наших знаний. Для того чтобы это уяснить, введем понятие *сообщения*. Вот два сообщения:

1. В данный момент вы читаете учебник.
2. Основным запоминающим элементом компьютера является релаксационная электронная схема, называемая триггером.

Первое сообщение не несет какой-либо *новизны*, не сообщает ничего нового. Во втором случае сообщение содержит новизну, так как раньше вы этого не знали. А если второе сообщение передать инженеру по компьютерной технике, будет ли оно содержать для него новизну? Конечно, нет. Для разных приемников сообщений одно и то же сообщение может содержать, а может и не содержать элемент новизны.

Если говорить проще, **информация** — это сообщение, содержащее элемент новизны для данного приемника (потребителя). Но раз есть приемник сообщений, то обязательно должен быть и источник, т.е. в общем виде имеем схему передачи информации, представленную на рис. 2.1.

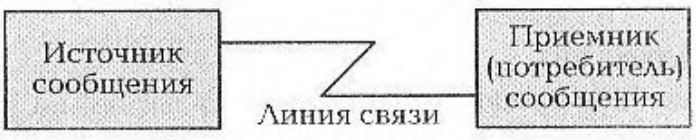


Рис. 2.1. Схема передачи сообщений (информации)

Вы читаете эту книгу — ее можно рассматривать как *источник сообщений*, а если то, что вы читаете, является для вас новым, то и как *источник информации*. Приемником или потребителем информации в этом примере является читатель. А линия связи в приведенном примере — это оптическая среда — воздух и глаза, т. е. то, что помогает читать.

Мы уже говорили, что одно и то же сообщение в зависимости от потребителя может как нести, так и не нести в себе информацию. Попробуем изобразить ситуацию знакомства (рис. 2.2). Источником сообщения, предложенного на схеме, является человек, имени которого вы не знаете, а приемниками, или, иначе говоря потребителями, соответственно мы. Раз сообщение передается голосом, то линией связи выступает воздух, благодаря акустическим свойствам которого мы, собственно, и слышим. Будет ли нести это сообщение информацию для нас, т. е. будет ли оно информативно? Конечно, ибо до данного сообщения вы не знали имени человека. Однако для родственников этого человека информация о его имени будет неинформативной.

Мы часто слышим фразу: «Только что мы получили новую информацию». Как вы думаете, правильна ли эта фраза? Уверены, что вы скажете — нет, неправильная. И действительно, информация — это не ботинки, которые могут быть и новыми, и старыми. Информация может либо быть, либо не быть. Третьего состояния быть не может. Другое дело, что она может являться своевременной или нет, подробной или краткой, правдивой или лживой, полной или неполной. Но это уже определяется ее *свойствами*: актуальностью, достоверностью, агрегированностью, мерой.

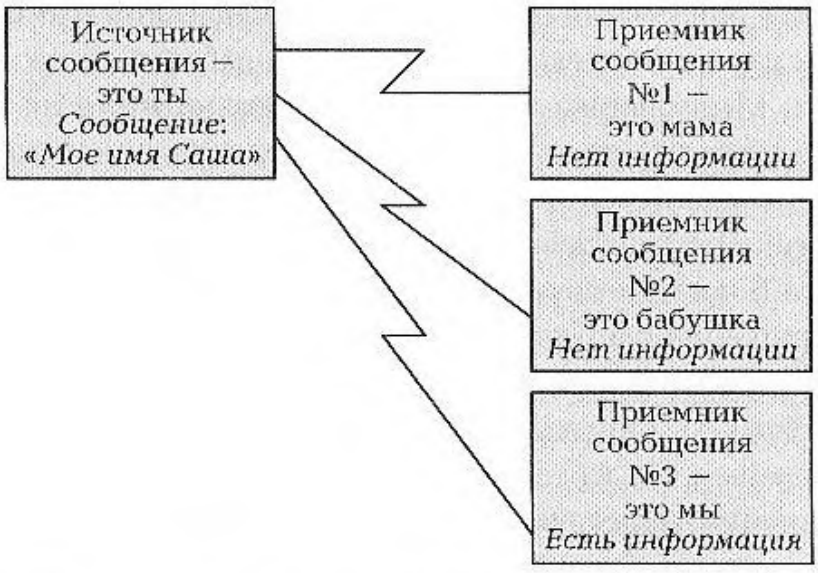


Рис. 2.2. Информативность сообщения для разных приемников

Актуальность (своевременность информации). Представьте себе, что полководец, командующий армией, накануне решительного сражения с неприятелем, расположившимся на другой стороне реки, послал несколько групп разведчиков с заданием раздобыть план расположения войск противника, его артиллерийских батарей, командного пункта, складов с боеприпасами, пунктов связи, т.е. диспозицию. Рассмотрим две ситуации: в первой самый отважный и самый находчивый разведчик за несколько часов до начала сражения принес копию секретной карты расположения войск противника. Используя эту актуальнейшую в данный момент информацию, командующий внес соответствующие коррективы в план боевых действий своих войск, значительно повысив тем самым вероятность достижения успеха в сражении. Во второй ситуации он так и не дождался ни одного разведчика до начала сражения, сражение все равно было выиграно. Но вот спустя два дня после его завершения явился один из разведчиков и все-таки принес карту с диспозицией войск противника перед боем. Актуальна ли эта информация, да и информация ли это вообще? Ведь в ходе боевых действий многое стало известно о противнике, о местонахождении его основных подразделений, их численности и огневой мощи. Тем не менее при явно неактуальном характере эта карта полезна, так как дает возможность проанализировать правильность принятых решений, увидеть свои ошибки и оценить соразмерность той цены, которой далась победа. Следовательно, даже неактуальная информация может быть полезной.

Достоверность. Информация, как мы понимаем, всегда относится к кому-то или чему-то. И вот то, насколько точно, правдиво она отображает состояние или процесс, происходящий с тем объектом, о котором идет речь, и называется ее достоверностью. Мы не говорим о специальном искажении информации, называемом дезинформацией, это делается для умышленного введения в заблуждение потребителя информации. Мы говорим о нормальной, умышленно не искажаемой информации, которая тоже может быть недостоверной из-за ошибок людей, сбоев в работе систем передачи и приема информации. Представим, что телеметрическая информация, передаваемая в центр управления полета с космического корабля, по каким-то причинам утратила свою достоверность. Это, как цепная реакция, вызовет принятие целого ряда решений, которые наверняка окажутся неправильными, или, если говорить научным языком, неадекватными ситуации, что неминуемо приведет к тяжелым последствиям. Вот почему вопросами качества хранения, передачи, приема информации занимаются коллективы ученых, решая вопросы использования специальных кодов, повышающих надежность передачи, разрабатывающих методы и схемы

контроля, системы дублирования и резервирования. Поэтому, принимая ту или иную информацию, нужно всегда иметь определенную степень уверенности в ее достоверности.

Агрегированность. Данное свойство представляет собой степень укрупнения или детализации информации. Представим, что некий ученик не был знаком с компьютерами. Начался урок информатики, и учитель, помогая себе указкой, сказал: «Перед вами компьютер. Вот это — системный блок, там находится то, что, собственно, определяет функции компьютера; вот это — монитор, он нужен для визуального взаимодействия с компьютером; это — принтер, он печатает тексты, таблицы, графики, рисунки; это — клавиатура, с ее помощью мы вводим информацию в компьютер; а это — манипулятор «мышь», он дополняет возможности клавиатуры, особенно при работе с современными программами. Вот, дети, и все, что вам нужно знать о компьютере...». Сообщил ли он информацию? Если, как мы условились, ученик впервые увидел компьютер, то, несомненно, сообщил, но она слишком укрупненная, очень общая, т.е. обладает очень высокой степенью агрегированности. На следующий урок пришел другой учитель. Он согласился, что информация о компьютере, которую вы получили, обладает очень высокой степенью агрегированности. «Конечно же, эта информация слишком общая, — сказал он. — Я вам расскажу о компьютере более подробно...». И начал рассказывать о транзисторах, которые составляют основу аппаратной части компьютеров. Он повел свой рассказ об электронной и дырочной проводимости полупроводников, о зонной теории Ферми, об инжекции, экстракции и рекомбинации, об эмиттере, базе и коллекторе, о диффузионной и барьерной емкости, о p—n-переходе и потенциальном барьере, о схемах включения транзисторов и многом другом. Внимательно слушая учителя, ученик мало что понимал, а в голове все время крутился вопрос: «А причем же здесь компьютер?». Если говорить строго, очень даже причем, но вопрос вполне правомерен, ибо учитель давал слишком детализированную, слишком подробную, обладающую очень малой степенью агрегированности информацию. Это уже другая крайность. Кстати, умение определять степень агрегированности информации для данного потребителя является в какой-то мере искусством.

Мера информации. На основании информации мы выносим суждения о предметах, явлениях и процессах, даем оценки людям и их поступкам, решаем (или не решаем) те или иные задачи, принимаем какие-то решения и т.п. Есть правило о том, что информация — это основа для принятия решения. А что нужно для того, чтобы решение было правильным? Конечно же, достаточное количество или мера той информации, на базе которой принимается решение.

Информация может быть представлена в различных видах — это тексты, числа, рисунки, звук, видео и т.д. Каждому виду информации в компьютере соответствует специальный формат, который можно определить по расширению файла, хранящего данную информацию, например:

- .doc — текстовая информация;
- .xls — табличная информация;
- .bmp — точечный рисунок;
- .jpg — векторный рисунок;
- .wav — звуковая информация;
- .mp3 — звуковая информация (музыка);
- .avi — видеоинформация и т.д.

Вопросы и задания

1. Что несет в себе информация для ее получателя?
2. Как связаны информация и сообщения?
3. Приведите примеры линий связи для передачи информации.
4. Расскажите о свойствах информации и приведите соответствующие примеры.
5. В каких видах может быть представлена информация? Определите по расширению файла, какого типа информация есть у вас на компьютере.

2.2. Информация и управление

После Второй мировой войны сформировалась новая наука — *кибернетика*, занимающаяся вопросами управления и обработки информации. Ее создателем был Норберт Винер, который утверждал, что понятия «информация» и «управление» неразрывно связаны между собой, ибо «информация — это основа управления». Само слово «кибернетика» далеко не новое, оно встречалось еще у древнего грека Платона и означало искусство управления кораблем. Известный французский физик Андре-Мари Ампер называл кибернетикой науку об управлении государством.

В настоящее время кибернетика занимается математическим описанием процессов управления в машинах, механизмах, сложных системах, в том числе и электронных, в живых организмах. Она же рассматривает общие законы получения, хранения, передачи и обработки информации, так что в развитии информатики кибернетика сыграла огромную роль. Основной объект исследования кибернетики — кибернетические системы, они рассматриваются абстрактно, вне зависимости от их материальной природы. К кибернетическим системам можно

отнести системы управления в технике, компьютеры, человеческий мозг, биологические популяции, человеческое общество и т.д. Теоретическое ядро кибернетики составляют теория алгоритмов, исследование операций, оптимальное управление, распознавание образов.

Рассмотрим процесс управления на примере термостата — автоматического устройства для поддержания заданной температуры (рис. 2.3). Он используется и в аппаратуре для тонких биохимических исследований, и даже в инкубаторах для выведения цыплят. Температура внутри устройства контролируется датчиком и постоянно сравнивается с заданной. Если температура понижается, то информация об этом в виде специального сигнала поступает в устройство, регулирующее электрический ток, к которому присоединен нагреватель, ток увеличивается и повышает температуру. Как только температура в устройстве достигнет заданной величины, датчик проинформирует об этом регулятор, который отключит или уменьшит ток.

В рассмотренной схеме можно выделить *прямую связь* — воздействие тока на температуру в инкубаторе через нагреватель, и *обратную связь* — информацию от датчика о температуре и, соответственно, команду (информацию) на усиление или уменьшение тока.

На этом простом примере мы познакомились с очень серьезным научным понятием — управлением с обратной связью, причем, как мы убедились, обратная связь — это, как правило, информация о ходе управляемого процесса, поступающая в элемент управления (в нашем примере — это регулятор тока).

Оказывается, что с точки зрения кибернетических принципов управления не важно, чем управлять, так как наука об управлении едина. Управление подразумевает исполнение следующих основных функций:

- *учет* — должно учитываться все, что характеризует управляемый объект (вспомните меру информации), в нашем примере — это температура;
- *контроль* — обычно контролируются показатели управляемого процесса, у нас эту функцию выполняет датчик температуры;



Рис. 2.3. Схема термостата

- *анализ* — например, сравнение фактической температуры с той, которая должна быть;
- *нормирование* — установление различных норм и нормативов, в нашем примере — это та температура, которую нужно поддерживать в инкубаторе;
- *планирование* — выработка плана действий по управлению процессом, у нас — это срок пребывания яиц в инкубаторе, по истечении которого должны вылупиться цыплята;
- *регулирование* — прямое воздействие на управляемый процесс на основании информации, полученной по обратной связи, в нашем примере это увеличение или уменьшение тока и, соответственно, температуры в инкубаторе;
- *организация* — подразумевает самые различные формы, в нашем случае — это организация схемы, позволяющей эффективно управлять процессом выведения цыплят в инкубаторе;
- *прогнозирование* — очень важная функция управления, говоря просто, это умение представлять себе, к чему могут привести последствия принимаемых решений или производимых действий. Что нужно спрогнозировать в нашей схеме? Мы должны представлять себе последствия технических неполадок, например, выхода из строя датчика, регулятора тока, устройства сравнения с заданной температурой, нарушения контакта в проводах и т. п. Нужно не только прогнозировать, но и принимать соответствующие меры на основании прогноза, в частности задублировать основные элементы схемы (например, установить два датчика), иметь в резерве ее основные элементы (например, регулятор тока, который в случае выхода из строя основного можно быстро включить в схему) и т. д.

Очевидно, что управляющий элемент осуществляет свои функции путем обработки и передачи информации. Иногда в отличие от нашего примера объемы такой информации поистине огромны, это можно наблюдать в управлении современными предприятиями и фирмами, летательными объектами, технологическими процессами, проектированием новой техники и т. д. Естественно, что человеку или простейшим техническим устройствам это не под силу. Необходим мощный инструмент, осуществляющий быструю и точную обработку информации. Именно таким инструментом является компьютер.

Однако было бы неправильным сводить огромный мир управления и информации только к компьютерам. Мы уже говорили о кибернетике и должны понимать, что кибернетические идеи возникли не на пустом месте. А что же тогда явилось основой для их возникновения?

Конечно же, живая природа. Как вы знаете, подсолнух в течение всего дня соцветием поворачивается к солнцу. Как вы думаете, а правомерно ли здесь говорить о наличии природной управляющей системы с обратной связью? Наверное, да. А могут ли быть управляющие системы без обратной связи? Конечно, могут. Они еще называются *разомкнутыми*, в таких системах информация как отклик на управляющий сигнал отсутствует. Везде, несмотря на совершенно различную природу объектов (биология, техника, социальная система), присутствуют одни и те же *информационно-управляющие закономерности*.

Действительно, везде есть управляющий элемент: канал (линия), по которому идет командная информация (прямая связь); датчик (датчики), передающий информацию о состоянии управляемого объекта управляющему элементу; канал (линия), по которому эта информация о состоянии управляемого объекта (обратная связь) приходит к нему и, соответственно, регулирующий элемент. Давайте попробуем обобщить все эти закономерности на схеме (рис. 2.4):

Выводы напрашиваются сами собой:

1. Управление в любых системах осуществляется на основании информации или, как говорил академик В.М.Глушков «информация — хлеб управления».

2. Структура систем управления с обратной связью (иногда их называют самоуправляемыми или замкнутыми) одинакова и не зависит от природы управляющих и управляемых элементов.

3. Роль информации во всех таких системах одинакова и не зависит от природы системы.

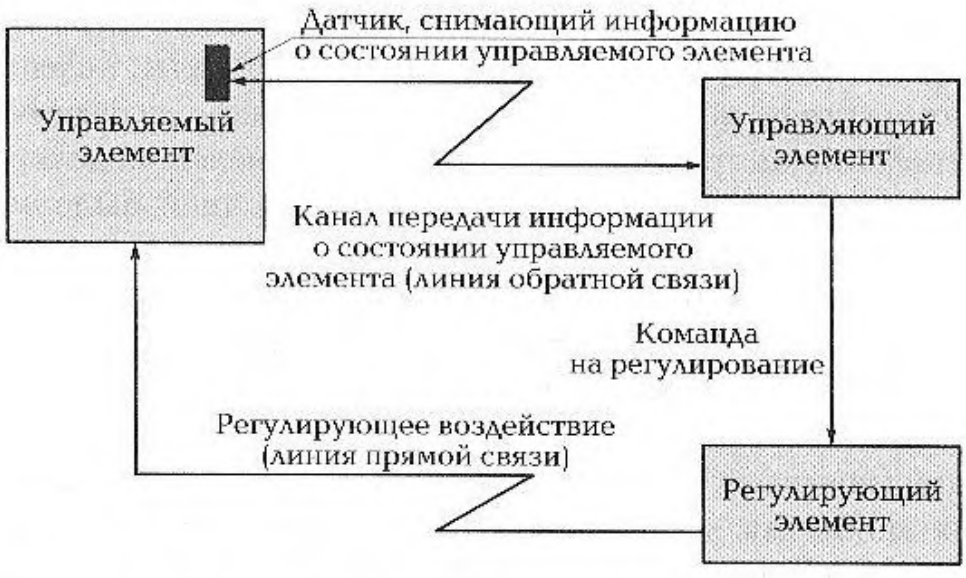


Рис. 2.4. Схема управления с обратной связью

Вопросы и задания

1. Поясните слова Норберта Винера «информация — это основа управления».
2. Опишите информационно-управляющие закономерности, используя схему, приведенную на рис. 2.4.
3. Объясните процесс управления на примере использования телевизионного пульта. Как с помощью ручного управления осуществляется устранение разомкнутости этой системы?
4. Постарайтесь описать работу станка с числовым программным управлением как замкнутую схему управления.
5. Кожный покров человека имеет специальные нервные окончания — рецепторы, которые можно рассматривать как датчики температуры в нашем примере с инкубатором, правда, их намного больше и они гораздо более чувствительны. Именно благодаря им мы чувствуем тепло или холод. Объясните на примере реакции человека на жару или холод систему управления терморегуляцией организма.

2.3. Информация и моделирование

Издревле люди пытались постичь суть окружающих их предметов и явлений с помощью *моделей*. Да и развитие человека начинается с моделей. Чем является большинство детских игрушек? Моделями объектов окружающего мира (домики, куклы, фигурки зверей и птиц, машинки и т. д.) или же моделями различных процессов (сборка, компоновка, поиск, выбор, последовательность и т. д.). Есть, как известно, игрушки, моделирующие цветовые, звуковые характеристики. Вам хорошо известно, как на уроках используются географические карты, чучела животных, различные геометрические фигуры, вы выполняете лабораторные работы по физике, ставите химические опыты и т. д. Все это позволяет вам лучше понять различные предметы, явления, процессы посредством моделирования. Таким образом, *моделирование* — это мощнейший инструмент познания. Первым видом моделирования, которое в определенном виде дошло и до наших дней, было *натурное моделирование*, которое применялось в строительстве, судоходстве и других областях. Суть его заключалась в том, что создавалась, как правило, уменьшенная модель, точно повторяющая формы и очертания моделируемого объекта. Вполне очевидно, что модель более доступна, чем сам объект. Воздействуя на эти модели такими факторами, как статическая нагрузка, ветер, вода, воздух, специалисты наблюдали за поведением объекта, делали определенные замеры интересовавших их параметров и с помощью формул пересчета с учетом масштабирования пытались спрогнозировать поведение реального объекта в аналогичных

условиях, т.е. фактически ими анализировались *информационные признаки моделей*, на основании которых и делались соответствующие выводы. Стоит отметить, что чисто натурное моделирование, сопровождающееся последующими ручными расчетами, было недостаточно эффективно и не всегда приводило к желаемым результатам. Поэтому с середины 1930-х гг. стало развиваться *полунатурное моделирование*, которое широко применяется и в наши дни. При таком процессе либо создавалась упомянутая выше модель, либо брался сам объект, и к нему присоединялись всевозможные датчики, необходимые для съема интересующей информации. С появлением цифровой вычислительной техники стали очевидными ее преимущества при проведении такого моделирования. Однако вся проблема заключалась в том, что информация с датчиков носила *аналоговый*, т.е. непрерывный, характер, а компьютер работает в *цифровом* (дискретном, т.е. прерывистом, импульсном) режиме. Возникла потребность в *аналого-цифровом* и *цифроаналоговом преобразовании*. Аналоговый сигнал — это непрерывный сигнал, его можно сравнить с тропинкой, ведущей в гору, — ноги идущего по ней человека могут стоять на любой высоте относительно основания горы, и, естественно, не выше самой горы (рис. 2.5, *а*). А теперь представьте, что построили лестницу, ведущую к вершине нашей горы. Здесь картина другая — ноги человека могут оказаться только на фиксированных высотах, определяемых расположением ступенек. Эти фиксированные значения называются *дискретами* или *отсчетами* (рис. 2.5, *б*).

На рис. 2.5, *в* слева показана просто тропинка (аналоговый сигнал), а справа — ступеньки в виде оснований стрелочек (цифровой сигнал). Причем если аналоговый сигнал непрерывен, то цифровой — дискре-

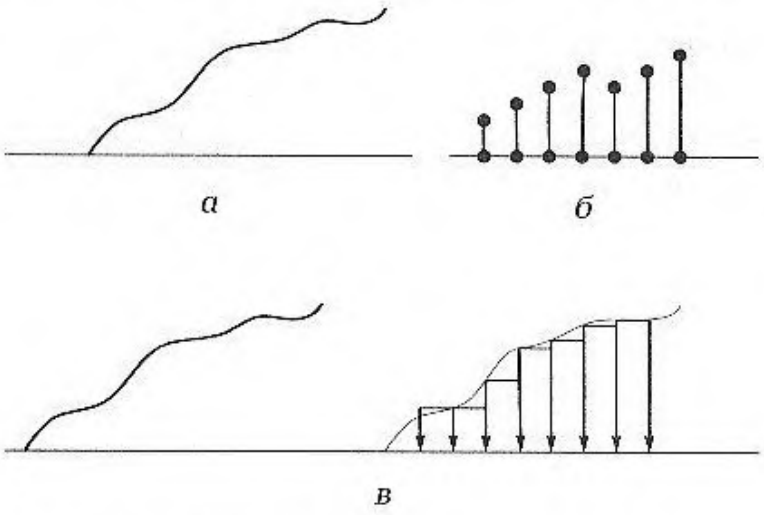


Рис. 2.5. Аналоговые и цифровые сигналы:

а — пример аналогового сигнала; *б* — пример цифрового (дискретного) сигнала; *в* — преобразование аналогового сигнала в цифровой

тен, т.е. квантован (квант — это порция) по времени, или, проще говоря, возникает в виде импульсов, имеющих различную амплитуду через определенные временны́е интервалы. Для преобразования цифровых сигналов в аналоговые используется *модуляция*. Она может быть амплитудной, амплитудно-частотной, фазовой, фазово-частотной и т.д. Наиболее понятной является амплитудная модуляция, суть которой состоит в том, что вершины дискретов (отсчетов) соединяются некой огибающей — и это будет уже непрерывный (аналоговый) сигнал. *Демодуляция* состоит в обратном — от плавной кривой перейти к совокупности дискретов (отсчетов).

Итак, с началом использования цифровой компьютерной техники в полунатурном моделировании его возможности значительно возросли, поскольку стали широко применяться различные математические методы обработки информации с датчиков, в частности прогнозирование развития процессов, сопоставление результатов, полученных математическим и эмпирическим (экспериментальным) путем.

И наконец, еще один, пожалуй, самый распространенный сегодня вид моделирования — информационное. *Информационные модели* — это описание объектов-оригиналов с помощью алгоритмов. Одним из самых распространенных видов этого моделирования является математическое. Суть его заключается в том, что некоторое явление, процесс или предмет описывается совокупностью математических закономерностей (формул, условий и т.д.), в результате чего образуется *математическая модель*, которая исследуется по специальным алгоритмам. Кстати, именно этим видом моделирования вы пользуетесь на уроках физики и химии при решении задач количественного характера.

Основа моделирования — информация. Давайте взглянем на проблему моделирования под несколько другим углом. Все ли задачи, которые перед нами ставит жизнь, являются четкими и ясными? В большинстве случаев конечно же нет. Поэтому одна из важнейших задач практической информатики — переход от нечетких задач к моделям, позволяющим решать эти задачи на компьютере. Поскольку большинство моделей содержат не только информационные признаки объектов, но и логические, такие модели принято считать *информационно-логическими* или *инфологическими*. Например, модели двух деталей одного механизма можно определить информационными признаками — геометрическая форма, размеры, материал, цвет и т.д., а вот процесс их взаимодействия в механизме — это уже логический признак, следовательно, только информационно-логическая модель даст полное представление о взаимодействии деталей. Процесс моделирования неразрывно связан с процессом *алгоритмизации*. Ре-

зультатом моделирования является набор величин и алгоритмов, полностью и однозначно описывающий объект, процесс или событие. Процесс моделирования всегда начинается с выделения существенной информации об объекте моделирования и представления этой информации некоторым набором алгоритмических величин. Именно в связи с моделированием в информатике появился так называемый *объектно-ориентированный подход*. Под *объектом* стали понимать единое целое, состоящее из информации и процедур обработки этой информации. Характерной чертой объектов является *инкапсуляция* (объединение) информации и алгоритмов (процедур) ее обработки, в результате чего и информация, и алгоритмические процедуры во многом теряют самостоятельное значение. Любой новый объект (производный от начального) может быть объявлен *потомком* ранее существующего объекта. При этом он наследует всю информацию и алгоритмы *объекта-родителя*, а также может содержать свою информацию и свои алгоритмы.

Следует отметить, что сама процедура моделирования является достаточно субъективной. Очевидно, что для одного и того же объекта можно сформировать различные модели, учитывающие разные факторы, определяющие свойства и поведение объекта. Собственно, правильный выбор таких факторов и определяет эффективность модели.

Принято различать графический (чертежи, графики, рисунки, эскизы), описательный (всевозможные словесные описания, например, словесный портрет человека), информационно-логический (об этом мы уже говорили) и математический виды представления моделей. Существуют модели, которые описывают моделируемый объект системой отношений, причем ряд параметров в этих отношениях находится во власти человека. Назначение этих моделей — найти такое сочетание параметров, при котором будет достигнут наилучший (оптимальный) результат из возможных.

Например, имеется несколько видов кормов для коров, каждый из которых имеет различное количество белков, жиров и углеводов, а также разную стоимость. Можно построить модель, которая обеспечит, с одной стороны, наиболее рациональное питание животных, а с другой — минимальные затраты. Такие модели называют *оптимизационными*. Наиболее широкое применение они нашли в экономике. Известный советский ученый, лауреат Нобелевской премии Л. В. Канторович разработал модель оптимального раскроя ткани, что стало большим достижением в развитии экономико-математического моделирования. Перечисленные модели характеризуются тем, что в них отсутствуют факторы, активно противодействующие достижению цели.

Однако подобные ситуации не такие уж редкие. Особенно ярко это проявляется в играх, где есть два и более участника, каждый из которых преследует свою цель. Существует раздел математики, который занимается моделированием таких ситуаций, он так и называется *теорией игр*, а соответствующие модели — *игровыми*. Кроме того, существует моделирование, позволяющее прогнозировать состояние сложных систем в зависимости от различных стратегий управления ими. Такое моделирование называется *имитационным* и является одним из самых сложных видов моделирования.

Вопросы и задания

1. Что лежит в основе моделирования?
2. Приведите примеры моделей из жизни.
3. Какую роль сыграли компьютеры в развитии процессов моделирования?
4. Поясните суть модуляции и демодуляции.
5. Укажите особенности информационных моделей.

2.4. Структурные информационные модели

Особое место в информатике занимают так называемые *структурные информационные модели*, которые определяют построение таких важных средств, как *базы данных* (структурированные хранилища информации) и, соответственно, *системы управления базами данных* (СУБД).

В тех случаях, когда необходимо воспринять, осмыслить и переработать большой объем информации, ее нужно *структурировать*, т. е. выделить в ней элементарные составляющие и их взаимосвязи. Можно провести аналогию со складом, в котором в беспорядке лежат различные товары. Очевидно, найти что-то нужное на таком складе трудно. Требуется наводить порядок. База данных — это своего рода склад информации, и ее структурирование служит как раз наведению порядка в нем.

Информационная структура представляет собой упорядоченную систему данных. Наиболее простыми информационными структурами являются списки, таблицы, схемы, графы. Простой пример табличного структурирования информации — школьное расписание уроков.

Основными структурными моделями являются иерархическая, сетевая и табличная.

Иерархическая модель представляется в виде дерева, где отдельные элементы объекта являются узлами, а стрелочки показывают связи между этими элементами (рис. 2.6).

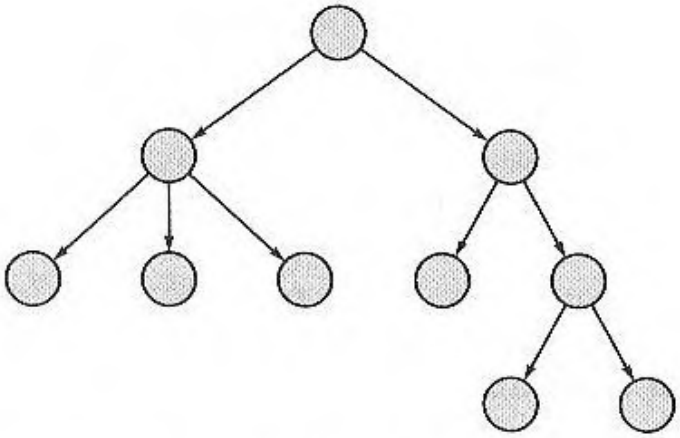


Рис. 2.6. Структура иерархической модели данных

Такая модель обладает некоторыми свойствами:

1. Иерархия начинается с верхнего узла. Каждый узел имеет характеристики (атрибуты), которые описывают моделируемый объект в данном узле.
2. Чем ниже уровень, тем выше «зависимость» узла.
3. Каждый узел имеет только одну связь с более высоким уровнем. Каждый узел может иметь несколько связей с «зависимыми» (более низкими) уровнями.
4. Доступ к любому элементу структуры осуществляется только через верхний узел по принципу «сверху-вниз».
5. Количество узлов не имеет ограничений.

В биологии весь животный мир рассматривается как иерархическая система (тип, класс, отряд, семейство, род, вид). В информатике используется иерархическая файловая система.

Рассмотрим процесс построения информационной модели, которая позволяет классифицировать современные компьютеры. Класс «Компьютеры» можно разделить на три подкласса: «Суперкомпьютеры», «Рабочие станции» и «Персональные компьютеры». Компьютеры, входящие в подкласс «Суперкомпьютеры», отличаются сверхвысокой производительностью и надежностью и используются в крупных научно-технических центрах для управления процессами в реальном масштабе времени.

Компьютеры, входящие в подкласс «Рабочие станции», обладают высокой производительностью и надежностью и используются в качестве серверов в локальных и телекоммуникационных сетях.

Компьютеры, входящие в подкласс «Персональные компьютеры», обладают средней производительностью и надежностью и используются в офисах и дома для работы с информацией.

Подкласс «Персональные компьютеры» делится в свою очередь на настольные, портативные и карманные.

Полученная информационная структура напоминает дерево, которое растет сверху вниз (именно поэтому такие информационные модели называют иногда древовидными). В структуре четко просматриваются три уровня, от первого, верхнего, имеющего один элемент компьютера, мы спускаемся до третьего, нижнего, имеющего три элемента: настольные, портативные (переносные) и карманные.

Основное отношение между уровнями состоит в том, что элементы нижнего уровня входят в состав одного из элементов верхнего уровня. В иерархической информационной модели объекты распределены по уровням, причем элементы нижнего уровня входят в состав одного из элементов более высокого уровня.

Ранее мы построили иерархическую статическую информационную модель, однако довольно часто иерархические структуры применяются и для описания процессов. Так, для описания исторического процесса смены поколений семьи используются динамические информационные модели в форме генеалогического древа.

В *сетевой модели* каждый узел может иметь любое количество связей с другими узлами без соблюдения какой бы то ни было иерархии (рис. 2.7).

Сетевые информационные модели применяются для отражения таких систем, в которых связь между элементами имеет сложную структуру. Например, различные части глобальной компьютерной сети Интернет (американская, европейская, российская и т. д.) связаны между собой высокоскоростными линиями связи. При этом какие-то части (американская) имеют прямые связи со всеми региональными частями, в то время как другие могут обмениваться информацией между собой только через американскую часть (например, российская и японская).

В *табличной* (*реляционной*) *модели* каждый объект моделируемой системы описывается в виде таблицы с набором атрибутов. *Атрибуты*, или *поля*, — это построчные ячейки таблицы. Взаимосвязь между таб-

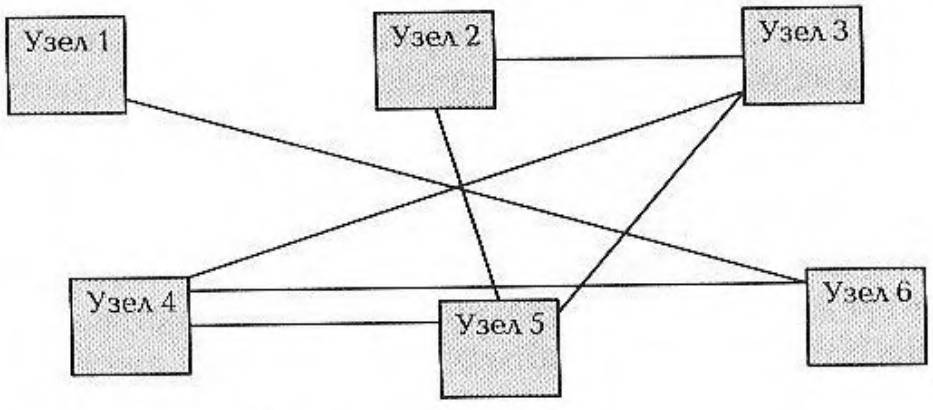


Рис. 2.7. Структура сетевой модели данных

Таблица 2.1

Код производителя	Название производителя	Адрес
...	...	...

Таблица 2.2

Код производителя	Код товара	Цена товара за единицу
...	...	...

Таблица 2.3

Код товара	Наименование товара
...	...

лицами описывается отношениями (реляциями) между полями. Например, компания, торгующая бытовой электроникой, наверняка имеет следующую информацию в разных таблицах (табл. 2.1–2.3).

Первую и вторую таблицы можно «связать» через поле **Код производителя**, вторую и третью — через поле **Код товара**.

Взаимосвязь между полями разных таблиц может иметь три вида:

1. «Один к одному». Одному элементу первого объекта соответствует только один элемент второго объекта (рис. 2.8). Например, конкретному человеку может соответствовать не более одного номера паспорта, а одному номеру паспорта — не более одного конкретного человека.
2. «Один ко многим». Одному элементу первого объекта может соответствовать несколько элементов второго объекта, а одному элементу второго объекта может соответствовать только один

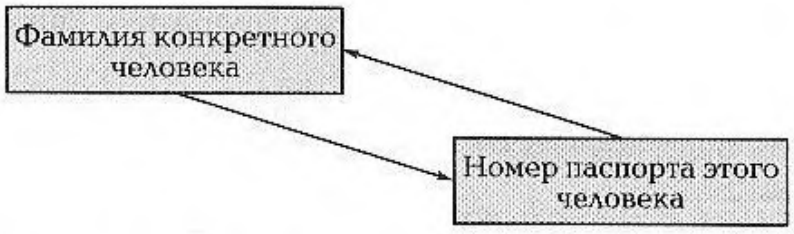


Рис. 2.8. Связь «один к одному»

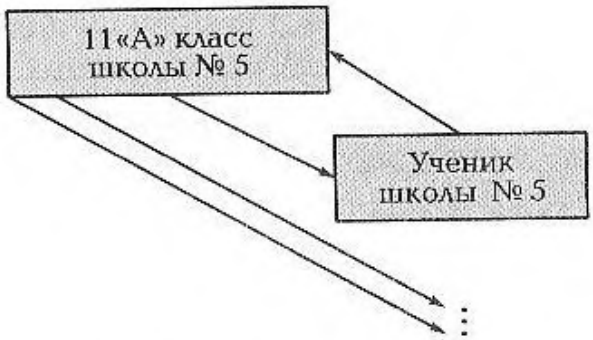


Рис. 2.9. Связь «один ко многим»



Рис. 2.10. Связь «многие ко многим»

элемент первого объекта (рис. 2.9). Например, в 11 «А» классе школы № 5 может учиться несколько учеников, а конкретный ученик школы № 5 может учиться не более чем в одном классе.

3. «Многие ко многим». Каждому элементу первого объекта может соответствовать множество элементов второго, и каждому элементу второго — множество элементов первого (рис. 2.10). Например, один предмет школьной программы могут изучать многие ученики, и один ученик может изучать многие предметы школьной программы.

Вопросы и задания

1. Что такое структурная информационная модель?
2. На примере файловой системы на вашем компьютере опишите иерархию ее уровней.
3. Опишите табличную модель данных на примере расписания занятий.
4. Нарисуйте иерархическую структуру «Компьютеры» для примера из параграфа.
5. Опишите сетевую модель на примере локальной компьютерной сети вашего учебного заведения.

П-СЭ **Проектное задание**

На примере табл. 2.1—2.2 предложите расстановку названий полей для разработки структуры БД «Расчет зарплаты». Используйте следующие ключевые слова: «ФИО», «паспортные данные», «адрес», «отдел», «должность», «размер зарплаты» в соответствии с должностью. Свяжите 1-ю и 2-ю таблицы через поле **ФИО**, а 2-ю и 3-ю через поле **Должность**.

2.5. Пример построения математической модели

Независимо от вида моделирования его основой всегда является информация. Изучение информационных процессов не является строгой прерогативой информатики, фактически любая наука в той или иной степени изучает информацию и информационные процессы (лингвистика — формы информационного общения людей, астрономия — визуальную или электронную информацию о строении Вселенной, социология — роль информации в обществе, биология — информационные процессы в живой природе и т.д.).

Можно еще раз сказать, что основная задача информатики в области моделирования — переход от неформальной, нечеткой задачи к четкой *формальной модели*. В качестве примера рассмотрим моделирование такого процесса, как полет тела, брошенного с некоторой начальной скоростью v_0 под углом α к горизонту.

Основные информационные характеристики процесса:

а) x, y — положение тела в пространстве, м;

б) v_x, v_y — скорость движения тела относительно осей X и Y, м/с;

в) a_x, a_y — ускорение тела относительно осей X и Y, м/с^2;

г) m — масса тела, кг.

Алгоритмической основой этой аналитической модели являются законы Ньютона. Единственная сила, действующая на тело, — сила тяжести $P = mg$. Следовательно, относительно оси Y процесс можно рассматривать как движение тела, брошенного вертикально вверх, а относительно оси X — как равномерное движение, т.е.

$$a_y = -g; \qquad a_x = 0.$$

Отсюда:

$$v_x(t) = v_{0x} = v_0 \cos\alpha; \qquad v_y(t) = v_{0y} - gt = v_0 \sin\alpha - gt.$$

Положение в пространстве в заданный момент времени t определяется следующими формулами:

$$x(t) = x_0 + v_{0x}t;\ y(t) = y_0 + v_{0y}t - \frac{gt^2}{2}.$$

Таким образом, математическая модель этого физического процесса будет иметь следующий вид:

$$a_x = 0;$$
$$a_y = -g;$$
$$v_x(t) = v_0 \cos\alpha;$$
$$v_y(t) = v_0 \sin\alpha - gt;$$
$$x(t) = x_0 + v_0 \cos\alpha t;$$
$$y(t) = y_0 + v_0 \sin\alpha t - gt^2/2;$$
$$t \in [0; T],$$ где T – момент падения тела.

На основании этой модели можно исследовать процесс. Например:

а) задав $Y = 0$ в уравнении движения по Y и решив уравнение относительно t, определить T — общее время падения тела;

б) полагая, что в самой верхней точке подъема скорость равна 0, и подставив это условие в уравнение для вертикальной скорости, найти время подъема в самую верхнюю точку траектории;

в) подставив полученное значение T в уравнение $x(t)$, вычислить максимальную дальность полета тела.

Таким же образом можно определить еще несколько параметров модели, например угол к горизонту, под которым бросают тело, при котором для одной и той же начальной скорости будет достигнута максимальная дальность полета; угол, под которым произойдет падение на землю, и т. д. Следовательно, можно попытаться формализовать состав и последовательность действий при моделировании процесса полета тела, брошенного под углом к горизонту, хотя, строго говоря, специфика моделируемых объектов внесет свои коррективы в алгоритмы моделирования.

П-ЕН Проектное задание

Опишите с использованием гармонических колебаний математическую модель колебаний маятника или работу колебательного контура. Используя полученную математическую модель, рассмотрите математическую модель системы биологических ритмов человека на сайте http://ru.wikipedia.org/wiki/Биоритм. Исследовать эту компьютерную модель (математическую модель, реализованную на компьютере) можно в онлайн-режиме на сайтах: http://www.ritms.ru/.

2.6. Единицы измерения информации

Вы уже знакомы с различными единицами измерения длины, массы, температуры и знаете, что единицей длины является метр, единицей силы — ньютон, единицей массы — килограмм и т.д. Мы недавно говорили о таком свойстве информации, как мера, следовательно, правомерна речь о *количестве информации*, а поскольку есть количество, то должны быть и единицы его измерения. Попробуем описать единицу измерения информации. Существует много явлений, которые могут иметь только два состояния. Например, выключатель бывает либо включен, либо выключен, дверь — открыта или закрыта, ток в проводнике — течет или не течет, свет — либо горит, либо не горит, дождь — идет или не идет. Таких примеров множество. Если обобщить сказанное, то эти состояния можно обозначить как «Да» или «Нет», а в цифровой форме «1» или «0», или двойной форме. Такая единица измерения двух состояний называется *битом*. При этом основная единица измерения информации в компьютере включает в себя 8 бит и называется *байт* (1 байт). Почему байт состоит из восьми бит? Обратите внимание, что 8 бит могут дать 256 различных комбинаций из нулей и единиц:

00000000
00000001
00000010
00000011
00000100
00000101
.....
.....
11111100
11111101
11111110
11111111

При этом 7 бит дают 128 комбинаций, а этого недостаточно, чтобы каждой комбинации поставить в соответствие буквы латинского и русского алфавитов и различные специальные символы. Заметим, что 9 бит дают 512 комбинаций, что избыточно.

Именно восьмибитный размер байта используют стандартные коды, «зашитые» в компьютер. Это кодировки ASCII (American Standard Code for Information Interchange — американский стандартный код для обмена информацией) и ANSI (American National Standards Institute — Американский национальный институт стандартов). Кста-

ти, определить количество комбинаций достаточно просто: если имеется N элементов, каждый из которых может иметь M состояний, то общее количество комбинаций составляет M^N. Например, имеется пять элементов, каждый из которых может иметь два состояния. Тогда $M = 2$, а $N = 5$. Следовательно, общее количество комбинаций $M^N = 2^5 = 32$.

Итак, с помощью одного байта можно описать один символ, букву, цифру. С учетом этого слово «весна» — это информация длиной в 5 байт. Для обозначения больших объемов информации используют производные единицы — *килобайт* (1 Кбайт), что составляет примерно 1 000 байт, или точнее 1 024 байт как 2 в степени 10. Существуют и большие меры: мегабайт, гигабайт, терабайт:

1 килобайт (1 Кбайт) = 1 024 байт;

1 мегабайт (1 Мбайт) = 1 024 килобайт = 1 024 · 1 024 байт;

1 гигабайт (1 Гбайт) = 1 024 мегабайт = 1 024 · 1 024 килобайт = = 1 024 · 1 024 · 1 024 байт;

1 терабайт (1 Тбайт) = 1 024 гигабайт.

Вопросы и задания

1. Обоснуйте выбор единиц измерения информации в компьютере — бит и байт. Как они связаны?
2. Сколько различных комбинаций единиц и нулей может быть записано в одном байте? Объясните, каким образом можно использовать эти комбинации для обозначения букв и цифр.
3. Назовите основные и производные единицы измерения информации.
4. Сколько байт в 2,6 Гбайт?
5. Что больше, 36 Кбайт или 0,000037 Гбайт? Обоснуйте свой ответ.

2.7. Системы счисления

С самого раннего детства нас учат считать, чтобы мы умели оценивать количество. Какая разница между числом и количеством? Одно и то же количество может быть выражено разными числами. Числа записываются с помощью цифр. Не следует путать понятия: цифра, число, количество. Использовать текстовую информацию позволяет алфавит, а количественную — системы счисления.

Система счисления — это система отображения любого числа с помощью ограниченного количества условных знаков, называемых цифрами.

Давайте докажем, что цифра — это условный знак для записи чисел. Возьмем число 8 136 547. Теперь представим себе, что цифра 3 обозначается как △, цифра 4 как □, цифра 5 как ○, остальные же цифры

остались прежними. При таких обозначениях цифр наше число будет выглядеть так:

$$8\,1\triangle\,6\,\bigcirc\,\square\,7.$$

Как вы думаете, изменилось ли количество, которое определяется этим числом? Конечно, нет. Изменился вид самого числа, да и то потому, что изменились условные знаки, называемые цифрами.

Трудно определить, сколько всего существует систем счисления. Скорее всего, бесконечное множество. Тем не менее их можно разделить на две группы: позиционные и непозиционные. В *позиционных системах счисления* вес цифры зависит от места (позиции), которую она занимает в числе.

Мы помним, что в записи числа используются позиции (разряды) единиц, десятков, сотен, тысяч, десятков тысяч, т. е. число можно представить в следующем виде:

$$35\,876 = 6\cdot 1 + 7\cdot 10 + 8\cdot 100 + 5\cdot 1\,000 + 3\cdot 10\,000\,.$$

Цифра 5, входящая в число 35 876, обозначает пять тысяч, потому что она находится именно в той позиции, в которой указывается количество тысяч, или, иными словами, именно нахождение в данной позиции определяет ее вес. В числе 68 952 тоже есть цифра 5, но ее вес, определяемый позицией в этом числе, составляет пять десятков.

В *непозиционных системах счисления* такой закономерности нет, т. е. вес цифры не зависит от ее позиции в числе. Классический пример непозиционной системы счисления — римская система, которая используется до сих пор, правда в основном для нумерации. Рассмотрим пример из двух римских чисел: $\overset{+1}{\text{II}}$ и $\overset{-1}{\text{IV}}$. Левая цифра в обоих числах — единица. В первом числе она имеет вес +1, а во втором числе –1.

Как вы понимаете, такие системы счисления имеют право на существование, но по причине отсутствия четких весовых закономерностей они неприменимы для компьютерной обработки информации.

Особый интерес из позиционных систем для нас представляют такие, веса которых являются членами геометрической прогрессии. Рассмотрим несколько рядов чисел:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1 024, 2 048 ...

1, 3, 9, 27, 81, 243, 729, 2 187, 6 561, 19 683, 59 045 ...

1, 4, 16, 256, 1024, 4 096, 16 384, 65 536, 262 144 ...

1, 5, 25, 125, 625, 3 125, 15 625, 78 125, 390 625 ...

Что общего в этих рядах чисел? Каждое следующее число в них получается из предыдущего путем умножения его на конкретное чис-

ло. В первом ряду это 2, во втором 3, в третьем 4, в четвертом 5 и т.д.

Такой ряд чисел называется *геометрической прогрессией*, сами числа ряда — это *члены геометрической прогрессии*, а то число, умножая на которое предыдущий (или n-й) член прогрессии, мы получаем последующий, или $(n + 1)$-й, является *знаменателем геометрической прогрессии*, обозначим его p.

Давайте теперь представим десятичное число 64 572 187 в весовом виде:

$$64\,572\,187 = 7 \cdot 1 + 8 \cdot 10 + 1 \cdot 100 + 2 \cdot 1\,000 + 7 \cdot 10\,000 + \\ + 5 \cdot 100\,000 + 4 \cdot 1\,000\,000 + 6 \cdot 10\,000\,000$$

и в виде таблицы по весам позиций и цифрам в этих позициях:

Вес позиции.....	1	10	100	1 000	10 000	100 000	1 000 000	10 0000 000
Цифра	7	8	1	2	7	5	4	6

Видно, что веса привычной нам десятичной системы счисления являются членами геометрической прогрессии, знаменателем такой прогрессии выступает число 10, т.е. $p = 10$, оно называется *основанием системы счисления*, а сама система называется *p-ричной системой счисления*. Запишем теперь число из таблицы с использованием степеней числа 10 — основания десятичной системы счисления:

$$64\,572\,187 = 7 \cdot 10^0 + 8 \cdot 10^1 + 1 \cdot 10^2 + 2 \cdot 10^3 + 7 \cdot 10^4 + \\ + 5 \cdot 10^5 + 4 \cdot 10^6 + 6 \cdot 10^7. \qquad (2.1)$$

Теперь попробуем записать представления числа по степеням основания p-ричной системы счисления в общем виде:

$$a_n p^n + a_{n-1} p^{n-1} + a_{n-2} p^{n-2} + \ldots + \\ + a_2 p^2 + a_1 p^1 + a_0 p^0, \qquad (2.2)$$

где a_i — цифра, находящаяся в позиции, имеющей вес $i = n, n - 1, n - 2, \ldots, 2, 1, 0$ соответственно; $n, n - 1, n - 2, \ldots, 2, 1, 0$ — веса позиций, или степень, в которую возводится p в данной позиции; p — основание системы счисления.

Но ведь число может иметь и дробную часть, а веса позиций в дробной части — числа отрицательные. Рассмотрим пример десятичного дробного числа 0,874562. Веса позиций, если идти от десятичной запятой вправо, таковы:

$$\frac{1}{10};\ \frac{1}{100};\ \frac{1}{1\,000};\ \frac{1}{1\,000};\ \frac{1}{100\,000};\ \frac{1}{1\,000\,000}.$$

Если использовать десятичные дроби, то веса будут выглядеть следующим образом:

$$0{,}1;\ 0{,}01;\ 0{,}001;\ 0{,}0001;\ 0{,}00001;\ 0{,}000001,$$

или то же самое с использованием отрицательных степеней числа 10:

$$10^{-1};\ 10^{-2};\ 10^{-3};\ 10^{-4};\ 10^{-5};\ 10^{-6}.$$

Значит, аналогично записи (2.1) предложенное дробное число можно представить так:

$$8 \cdot 10^{-1} + 7 \cdot 10^{-2} + 4 \cdot 10^{-3} + 5 \cdot 10^{-4} + 6 \cdot 10^{-5} + 2 \cdot 10^{-6}.$$

Следовательно, выражение (2.2) для чисел, имеющих как целую, так и дробную часть, примет следующий вид:

$$a_n p^n + a_{n-1} p^{n-1} + a_{n-2} p^{n-2} + \ldots + a_2 p^2 + a_1 p^1 + a_0 p^0 + a_{-1} p^{-1} + \\ + a_{-2} p^{-2} + \ldots + a_{-m} p^{-m}. \qquad (2.3)$$

Мы говорили о том, что любая система счисления использует ограниченное число условных знаков — цифр. Оказывается, что количество этих знаков равно основанию системы p. И действительно, в десятичной системе их десять, это 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. А сколько их должно быть, например, в шестеричной системе счисления, т.е. когда $p = 6$. Цифр будет шесть, это 0, 1, 2, 3, 4, 5. А в восьмеричной? Восемь, это 0, 1, 2, 3, 4, 5, 6, 7.

Сформулируем *правило*: количество цифр в p-ричной системе счисления равно p, причем это цифры от 0 до $(p - 1)$.

Как быть, если основание системы больше 10, например 12 или 16? В этом случае на помощь приходят буквы латинского алфавита: A, B, C, D и т.д. Правда в этом случае они называются уже не буквами, а цифрами. Итак, основание системы счисления может быть любым, все системы равноправны, но тем не менее мы используем десятичную систему счисления. Причина проста: на руках 10 пальцев — это, наверное, и есть наш первый «вычислитель».

Особо мы будем еще рассматривать двоичную систему счисления, поскольку она более удобна для использования в компьютерах. Мы уже говорили, что один двоичный разряд, который может содержать либо «0», либо «1», иначе называется битом, т.е. порция информации в 1 бит это и есть либо «0», либо «1».

Основным элементом, который хранит 1 бит информации в компьютере, является *триггер*. Для хранения нескольких бит информации используются столько элементов, сколько бит надо хранить. Как правило, эти элементы и есть триггеры, в этом случае они образуют интегральную схему, которая называется *регистром*. Если попытаться

дать более строгое определение, то регистр — это совокупность элементов, которые могут принимать, хранить и выдавать информацию в компьютере. Как мы узнаем позже, регистры играют очень большую роль в работе микропроцессора компьютера и других его частей.

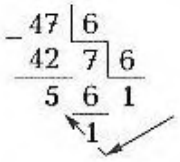
Вопросы и задания

1. Что называется системой счисления?
2. Приведите доказательства того, что цифра не более чем условный знак. Какие различия между понятиями: цифра, число, количество? Обоснуйте свое мнение.
3. Почему десятичная система счисления наиболее привычна для нас?
4. Сколько цифр должно быть в семеричной системе счисления? Может ли цифра 8 входить в состав восьмеричной системы счисления?
5. Что такое вес позиции в системе счисления? Имеется десятичное число 324 512. Какой вес имеет каждая позиция?

2.8. Модель перевода чисел из одной системы счисления в другую

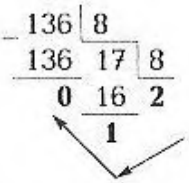
Перевод чисел из одной системы счисления в другую можно проводить по общему правилу. Для того чтобы целое число из системы счисления с основанием p_1 перевести в систему счисления с основанием p_2, нужно последовательно делить его на p_2, выраженное в системе с основанием p_1. Результат — остатки от промежуточных делений, записанные справа налево.

Например, нужно десятичное число 47 перевести в шестиричную систему счисления:



Таким образом, $47_{10} = 115_6$.

Или десятичное число 136 переведем в восьмеричную систему счисления:

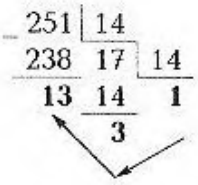


Таким образом, $136_{10} = 210_8$.

Теперь попробуем перевести десятичное число в систему счисления с основанием больше 10, например в 14-ричную. Очевидно, что при

последовательном делении на 14 у нас могут получаться промежуточные остатки от 0 до 13, и привычных цифр нам не хватит. Условимся о следующих обозначениях: 10 — A; 11 — B; 12 — C; 13 — D.

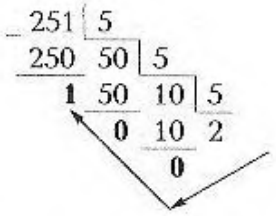
Переведем десятичное число 251 в 14-ричную систему счисления:



С учетом нашей договоренности о «новых» цифрах имеем:

$$251_{10} = 13D_{14}.$$

Интересно, как будет выглядеть десятичное число 251, например в пятеричной системе счисления:



Таким образом, $251_{10} = 13D_{14} = 2001_5$.

На этом примере мы хотим вернуться к выводу, что число и количество — понятия разные. Теперь можно объяснить это, сказав, что в различных системах счисления одно и то же количество выражается разными числами.

Чтобы убедиться, что перевод выполнен правильно, необходимо сделать обратный перевод. Это делается путем «расписывания» числа по степеням основания в соответствии с правилом (2.2).

Переведем число 13D из 14-ричной системы в десятичную:

$$13D_{14} = D \cdot 14^0 + 3 \cdot 14^1 + 1 \cdot 14^2.$$

С учетом наших договоренностей об обозначении цифр латинскими буквами получим:

$$13D_{14} = 13 \cdot 14^0 + 3 \cdot 14^1 + 1 \cdot 14^2 = 13 \cdot 1 + 3 \cdot 14 + 1 \cdot 196 = \\ = 13 + 42 + 196 = 251_{10}.$$

Переведем число 2001 из пятиричной системы в десятичную:

$$2001_5 = 1 \cdot 5^0 + 0 \cdot 5^1 + 0 \cdot 5^2 + 2 \cdot 5^3 = \\ = 1 \cdot 1 + 0 \cdot 5 + 0 \cdot 25 + 2 \cdot 125 = 1 + 0 + 0 + 250 = 251_{10}.$$

Удобнее «расписывать» числа по степеням, начиная с младших разрядов.

Над числами, представленными в различных системах счисления, можно выполнять *арифметические операции* так же, как это мы делаем с десятичными числами. Сложим в столбик два пятиричных числа:

$$\begin{array}{r} 24324 \\ +\ 43243 \\ \hline 123122 \end{array}$$

Почему получился такой результат? Выпишем числа по порядку в пятеричной системе счисления:

0, 1, 2, 3, 4, 10, 11, 12, 13, 14, 20, 21, 22, 23, 24, 30, 31, 32, 33, 34, 40, 41, 42, 43, 44, 100, 101, 102, 103, 104, 110, 111, 112, 113, 114, 120 ...

Вспомним, что в этой системе есть только пять цифр, это 0, 1, 2, 3, 4, и после числа 4 следует 10, после 14 — 20, после 44 — 100, после 444 — 1000 и т. д., т. е. осуществляется переход в следующую позицию (разряд).

А теперь сложим два 15-ричных числа. Цифры этой системы таковы:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E.

$$\begin{array}{r} 1CD93 \\ +\ 5E6A4 \\ \hline 71547 \end{array}$$

Вопросы и задания

1. Имеется 16-ричное дробное число 1A34C,2B9. Переведите его в десятичную систему счисления. Используйте правило (2.3).
2. Какое число отображает большее количество — пятеричное 4123 или 16-ричное 21B? Обоснуйте свой ответ.
3. Найдите сумму двух девятеричных чисел 2854,63258 и 8745,3775.
4. Объясните полученный в последнем примере параграфа результат, используя числа, выписанные по порядку в 15-ричной системе счисления: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 40, 41, 42 ...
5. Выпишите по порядку числа в восьмеричной системе счисления. Представьте восьмеричное число 7 в двоичной системе счисления. Что можно сказать о количестве триггеров в регистре для восьмеричных кодов в компьютере? Почему они называются триадами?

Проектное задание

Реферат «Двоичное кодирование и компьютер» необходимо выполнить на основе нескольких тем курса. Подготовьте раздел о видах позиционных систем — почему именно позиционные системы нашли широкое применение в компьютерной технике? Обоснуйте пары понятий бит и триггер, байт и регистр. Используйте задание № 5 к параграфу в качестве примера.

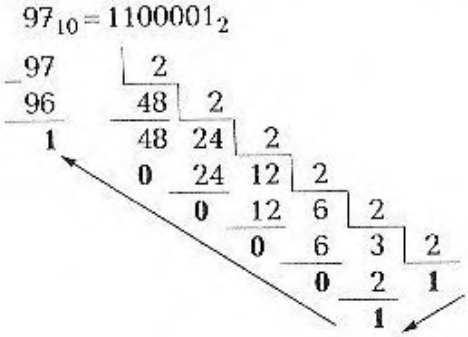
2.9. Двоичная и шестнадцатеричная системы счисления как модель представления чисел в компьютере

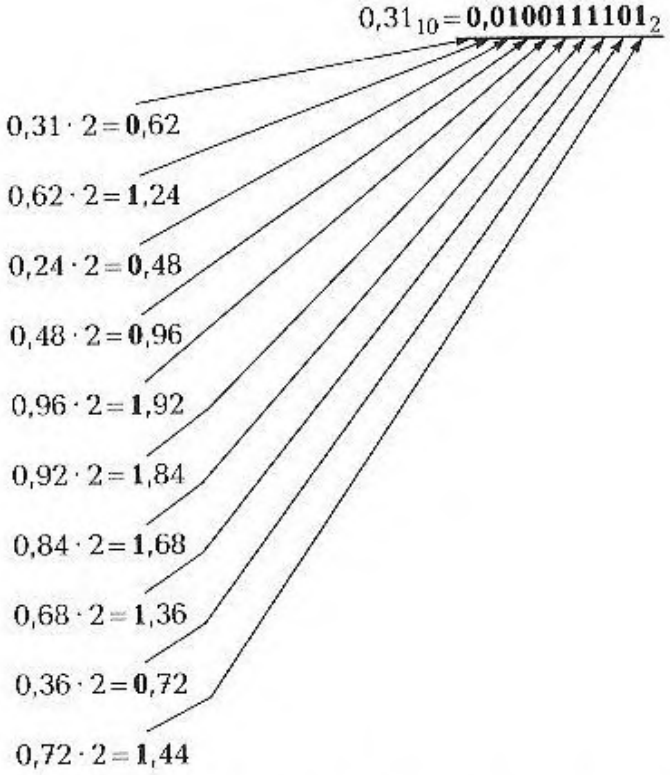
Двоичная и 16-ричная системы счисления принципиально ничем не отличаются от других систем, которые мы рассматривали. Тем не менее они представляют особый интерес потому, что удобны для использования в компьютерах. Сначала о двоичной системе. Вы уже понимаете, что исходя из общего правила, основанием этой системы является число 2, т.е. $p = 2$. В ней должно быть p цифр, т.е. две, и цифры эти от 0 до $(p - 1)$, т.е. 0 и 1. Основное качество, которым должен обладать любой «обработчик» информации, будь то человек или компьютер, — это способность запоминать. Информацию, подлежащую обработке, нужно помнить или, иначе, хранить. Вы наверняка помните телефоны друзей, имена и фамилии любимых спортсменов или артистов, даты рождения родных и близких людей и многое другое. В нужный момент вы «извлекаете» необходимую информацию из памяти и используете ее. Наука предполагает, что человеческая память состоит из отдельных элементов — нейронов, которые, к сожалению, могут и «ломаться», тогда наступает тяжелая болезнь — утрата памяти, врачи называют ее амнезией. Память компьютеров тоже состоит из отдельных элементов, своеобразных «компьютерных нейронов». Но эти элементы искусственные, т.е. являются творением человека.

Сосредоточимся пока на запоминании числовой информации. Казалось бы, чего проще: нужно использовать привычную нам десятичную систему. Не получается. Судите сами — в десятичной системе 10 цифр, значит, каждый элемент памяти должен иметь 10 различных положений, и не просто положений, а устойчивых, фиксированных положений. Ранее мы обсудили элементы, имеющие два устойчивых положения, которые можно описать двумя состояниями: «да» или «нет». Каждое из этих двух состояний можно обозначить, одно как «0», другое — как «1». Но это не что иное, как двоичная система счисления. Вот он и «компьютерный нейрон». Именно это обстоятельство и является причиной того, что двоичная система занимает особое место среди других систем счисления, с ее помощью наиболее просто описать работу искусственного элемента памяти компьютера.

Двоичная система счисления строится по тем же правилам, что и остальные позиционные системы. Возьмем десятичное число, например 97,31, и представим его в двоичной системе счисления, при

этом в смешанном числе целая и дробная части переводятся по-разному: целая — путем последовательного деления на основание системы, в которую осуществляется перевод, а дробная — путем последовательного умножения. Результат же записывается совместно, через запятую, в данном случае это двоичная запятая:





Как вы можете замстить, мы выбрали точность перевода дробной части до десятого знака после двоичной запятой. На каждом шаге полученное значение целой части используется в «сборке» дробной части двоичного числа.

Таким образом,

$$97,31_{10} = 1100001,0100111101_2.$$

Теперь составим таблицу, в верхней строке которой будут веса позиций, а в нижней — цифры двоичного числа, находящиеся в этих позициях (табл. 2.4).

Таблица 2.4

Целая часть							Дробная часть									
2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}
1	1	0	0	0	0	1	0	1	0	0	1	1	1	1	0	1

Выполним обратный перевод целой части, используя приведенную таблицу:

$$1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 =$$
$$= 64 + 32 + 0 + 0 + 0 + 0 + 1 = 97.$$

Теперь аналогично переведем дробную часть:

$$0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 0 \cdot 2^{-4} + 1 \cdot 2^{-5} + 1 \cdot 2^{-6} + 1 \cdot 2^{-7} + 1 \cdot 2^{-8} +$$
$$+ 0 \cdot 2^{-9} + 1 \cdot 2^{-10} = 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + 0 \cdot \frac{1}{8} + 0 \cdot \frac{1}{16} + 1 \cdot \frac{1}{32} + 1 \cdot \frac{1}{64} +$$
$$+ 1 \cdot \frac{1}{128} + 1 \cdot \frac{1}{256} + 0 \cdot \frac{1}{512} + 1 \cdot \frac{1}{1024} =$$
$$= \frac{1}{4} \cdot 256 + \frac{1}{32} \cdot 32 + \frac{1}{64} \cdot 64 + \frac{1}{128} \cdot 8 + \frac{1}{256} \cdot 4 + \frac{1}{1024} \cdot 1 =$$
$$= \frac{256 + 32 + 16 + 8 + 4 + 1}{1024} = \frac{317}{1024} = 0{,}30957.$$

Определим погрешность перевода дробной части:

$$\Pi = 0{,}31 - 0{,}30957 = 0{,}00043.$$

Таким образом:

$$1100001{,}0100111101_2 = 97{,}30957_{10}.$$

Известно, что числа бывают отрицательными и положительными, а еще есть ноль, который знака не имеет. В двоичной системе минус обозначается как «1», а плюс как «0». Вы спросите: а как же ноль, он ведь не имеет знака? В компьютерах, которые используют двоичную систему счисления, ноль — это положительное число.

Прежде чем приступить к рассмотрению 16-ричной системы счисления, поговорим немного о комбинаторике — одном из разделов математики. Знание одного из правил комбинаторики очень пригодится в дальнейшем. Вот это *правило*.

Если имеется n элементов, каждый из которых может иметь M состояний, то K — общее количество комбинаций этих элементов — определяется как

$$K = M^n.$$

Давайте применим это правило к двоичным комбинациям. Очевидно, что для двоичной системы счисления $M = 2$ всегда, ибо каждая двоичная позиция в числе может иметь только два состояния: «0» и «1» (такие двоичные позиции еще называют двоичными разрядами), а n есть не что иное, как количество рассматриваемых двоичных разрядов. Например, как подсчитать, сколько комбинаций могут содержать пять двоичных разрядов? В соответствии с рассмотренным правилом $M = 2$, а $n = 5$, значит количество комбинаций

$$K = 2^5 = 32.$$

Ранее мы говорили о том, что основной единицей измерения количества информации является байт, а он в свою очередь состоит из восьми бит, т.е. из восьми *двоичных разрядов*. Структура байта имеет следующий вид:



Разделим эту структуру на две части, тогда в каждой будет по четыре бита. Четыре бита могут дать 16 комбинаций (2 в степени 4). Вот эти комбинации:

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Таким образом, система счисления, которая здесь описана, — 16-ричная. Можно сделать вывод, что содержимое байта может быть представлено двумя 16-ричными цифрами, или иначе *двумя тетрадами*. А это, конечно, намного удобнее для восприятия, чем восемь двоичных нулей и (или) единиц. Возьмем десятичное число, например

87,78, и переведем его в 16-ричную систему счисления по известным нам правилам:

$$\begin{array}{r|l} 87 & 16 \\ \underline{80} & 5 \\ 7 & \end{array}$$

$$0{,}78 \cdot 16 = 12{,}48;$$
$$0{,}48 \cdot 16 = 7{,}68.$$

Таким образом, $87{,}78_{10} = 57{,}C7_{16}$.

Делаем вывод: для представления этого числа в компьютере необходимо по меньшей мере два байта, или четыре тетрады (в записи числа они выделены подчеркиванием).

Теперь представим 16-ричное число 57,C7 в двоичном виде, заменив каждую 16-ричную цифру тетрадой из четырех двоичных разрядов. При этом старшие нули в первой цифре целой части можно отбросить:

$$57{,}C7_{16} = \underline{0101}\,\underline{0111},\underline{1010}\,\underline{0111}_2 = \underline{101}\,\underline{0111},\underline{1100}\,\underline{0111}_2.$$

Теперь переведем это число в десятичную систему. Перевод целой части удобнее производить, начиная с младших разрядов:

$$1\cdot 2^0 + 1\cdot 2^1 + 1\cdot 2^2 + 0\cdot 2^3 + 1\cdot 2^4 + 0\cdot 2^5 + 1\cdot 2^6 = 1+2+4+16+64 = \underline{87};$$

$$1\cdot 2^{-1} + 1\cdot 2^{-2} + 0\cdot 2^{-3} + 0\cdot 2^{-4} + 0\cdot 2^{-5} + 1\cdot 2^{-6} + 1\cdot 2^{-7} + 1\cdot 2^{-8} =$$

$$= \frac{1}{2}\cdot 128 + \frac{1}{4}\cdot 64 + \frac{1}{64}\cdot 4 + \frac{1}{128}\cdot 2 + \frac{1}{256}\cdot 1 = \frac{128+64+4+2+1}{256} = \frac{199}{256} = \underline{0{,}777}.$$

Таким образом,

$$1010111{,}11000111_2 = 87_{10} + 0{,}78_{10} = 87{,}78_{10}.$$

С учетом округления дробной части мы получили исходное десятичное число. Сделаем еще один *вывод*. Переводить числа из десятичной системы в двоичную удобней, если использовать схему перевода из десятичной в 16-ричную систему, а затем в двоичную с помощью тетрад, т.е. применять 16-ричную систему как промежуточную при переводе.

Вопросы и задания

1. Переведите 16-ричное число 3AC9 в двоичную систему, а двоичное число 10001110101 — в 16-ричную, используя тетрады. Разбиение на тетрады следует проводить слева направо в целой части и справа налево в дробной части двоичного числа.

2. Один переключатель имеет три положения, а таких переключателей установлено девять. Сколько комбинаций возможно?
3. Переведите десятичное число 79,37 в двоичную систему непосредственно и с промежуточным переводом в 16-ричную систему. Какой способ проще?
4. Переведите 16-ричное число 2A7,E41 в десятичную систему непосредственно и через двоичную систему.
5. Сколько комбинаций могут содержать 14 двоичных разрядов?

2.10. Двоично-кодированные системы

В двоично-кодированных системах счисления каждая цифра числа записывается двоичными числами, т.е. в двоичную систему переводится не само число, а его цифры, или, иными словами, они кодируются в двоичном виде. Для примера возьмем десятичное число 135.

В двоично-кодированном виде с помощью тетрад оно будет иметь вид

$$0001\ 0011\ 0101.$$

Но это не значит, что обратный перевод позволит получить исходное число, если вы переведете полученное двоичное число не по цифрам, а целиком. Действительно,

$$000100110101_2 = 1 + 4 + 32 + 64 + 256 = 357_{10}.$$

Обратите внимание, что для записи одной десятичной цифры мы использовали тетраду — четыре двоичных разряда. Как мы уже знаем, десятичные цифры от 0 до 9 в двоичной системе будут иметь вид от 0000 до 1001, т.е. тетрада используется не полностью. Другое дело — такие двоично-кодированные системы, цифры которых кодируются всеми комбинациями данного количества двоичных разрядов. Например, четверичная система (цифры от 0 до 3) — двумя двоичными разрядами (диадой): 00, 01, 10 и 11; а восьмеричная (цифры от 0 до 7) — тремя (триадой): от 000 до 111; 16-ричная — четырьмя (тетрадой); 32-я — пятью (пентадой) и т.д. Здесь такой подход, который привел нас к неудаче при работе с двоично-десятичной системой, оправдан. Давайте проверим. Например, мы имеем двоичное число 11010101,101101. Если мы захотим перевести его в четверичную систему, то выделим диады 11 01 01 01, 10 11 01, что составит $3111,231_4$. Тогда это число при переводе в десятичную систему даст следующий результат:

$$3111,231_4 = 1\cdot4^0 + 1\cdot4^1 + 1\cdot4^2 + 3\cdot4^3 + 2\cdot4^{-1} + 3\cdot4^{-2} + 1\cdot4^{-3} =$$
$$= 1 + 4 + 16 + 192 + 0,5 + 0,1875 + 0,0156 = 213,7031_{10}.$$

Для перевода этого же двоичного числа в восьмеричную систему распишем его по триадам, тогда

$$011\ 010\ 101{,}101\ 101 = 325{,}55_8.$$

Проверим переводом в десятичную систему:

$$325{,}55_8 = 5\cdot 8^0 + 2\cdot 8^1 + 3\cdot 8^2 + 5\cdot 8^{-1} + 5\cdot 8^{-2} =$$
$$= 5 + 16 + 192 + 0{,}625 + 0{,}0781 = 213{,}7031_{10}.$$

Теперь переведем наше двоичное число в 16-ричную систему, для чего распишем его по тетрадам:

$$1101\ 0101{,}1011\ 0100 = D5{,}B4_{16}.$$

Для проверки переведем полученный результат в десятичную систему:

$$D5{,}B4_{16} = 5\cdot 16^0 + 13\cdot 16^1 + 11\cdot 16^{-1} + 4\cdot 16^{-2} =$$
$$= 5 + 208 + 0{,}6875 + 0{,}0156 = 213{,}7031_{10}.$$

Для двоично-кодированных систем, каждая цифра которых представлена полным набором комбинаций двоичных разрядов, закономерности представления и перевода вполне очевидны и просты.

? Вопросы и задания

1. Назовите двоично-кодированные системы, цифры которых кодируются всеми комбинациями данного количества двоичных разрядов.
2. Как связаны двоичные коды для представления чисел в 8- и 16-ричных системах счисления с помощью триад и тетрад?
3. Как называется элемент, который хранит 1 бит информации в компьютерах в качестве основного носителя информации?
4. Что называется регистром?
5. Что такое диада, триада и тетрада? Сколько двоичных разрядов они включают в себя?

П-Т Проектное задание

Подготовьте раздел «Почему именно двоичная система счисления нашла широкое применение в компьютерной технике?» для реферата «Двоичное кодирование и компьютер».

2.11. Кодирование информации

Как вы, наверное, уже знаете, современные компьютерные системы уже давно не работают только с числовой информацией, как это было на начальном этапе их развития. Они наряду с числами могут обра-

батывать и буквы, и слова, причем на различных языках и, так называемые специальные символы, к которым относят следующие знаки: . , ! ? « # $ % ^ & * ( ) _ – = + \ * > < и др. Очевидно, что любое математическое выражение или строка букв состоит из этих отдельных элементов — букв, цифр, знаков и т.д. Будем в дальнейшем называть их *символами*. Для представления такой информации в компьютерах создают специальные коды, которые так и называют — *символьные коды*. Это очень существенный момент, не зря еще в 1964 г. в нашей стране был выпущен посвященный этим вопросам государственный общесоюзный стандарт — ГОСТ 10859–64. В последние 15–20 лет эти вопросы определяются международными стандартами. Например, большие вычислительные машины типа IBM 360 и IBM 370 и их аналоги, которые доминировали в мире в 1970–1980 гг., использовали специальный код EBCDIC (Extended Binary Coded Decimal Interchange Code — расширенный двоично-десятичный код обмена информацией), у нас его аналогом был код ДКОИ-8 (двоичный код для обмена информацией, 8 бит). Код ASCII применяется в ПК, совместимых с IBM, работающих под управлением операционной системы MS DOS (Microsoft Disk Operating System — дисковая ОС). Сейчас, как вы наверняка знаете, в основном используется ОС Microsoft Windows. Она применяет так называемую ANSI-кодировку. Но эта кодировка ориентирована на английский язык и не содержит символов кириллицы (русских букв), поэтому разработчик Windows — американская компания *Microsoft* — создала русскую версию ANSI-кодировки, которая приведена далее. Наша задача — не заучивать эти многочисленные символы, а понять механизм формирования кодов, который, кстати, практически одинаков во всех символьных компьютерных кодах.

Вы уже знаете, что в компьютере каждый символ представлен в виде байта, состоящего из восьми двоичных разрядов, называемых битами. Вы помните также, что содержимым бита может быть либо «0», либо «1», а также то, что 8 бит могут дать 256 комбинаций из «0» и «1». Вы также знаете, что 4 бит, представляющие тетраду байта, «свернуты» в 16-ричную цифру. Если вы внимательно посмотрите на приведенную таблицу русской версии ANSI-кодировки (рис. 2.11), то увидите, что каждая строка и столбец начинаются с 16-ричной цифры и соответствующей битовой комбинации, а на пересечении строк и столбцов в соответствующих клетках находятся кодируемые символы. Возьмем, например, знак «+». В строке стоит 16-ричная цифра 2 и соответствующая ей кодовая комбинация — тетрада 0010, а в столбце — 16-ричная цифра B и соответствующая ей кодовая комбинация 1011. Следовательно, знак «+» в компьютере представлен кодом

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0001	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	0010	32	33 !	34 "	35 #	36 $	37 %	38 &	39 '	40 (	41)	42 *	43 +	44 ,	45 −	46 .	47 /
3	0011	48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7	56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
4	0100	64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G	72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
5	0101	80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W	88 X	89 Y	90 Z	91 [	92 \	93]	94 ^	95 _
6	0110	96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g	104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
7	0111	112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w	120 x	121 y	122 z	123 {	124 |	125 }	126 ~	127 □
8	1000	128 Ђ	129 Ѓ	130 ‚	131 ѓ	132 „	133 …	134 †	135 ‡	136 €	137 ‰	138 Љ	139 ‹	140 Њ	141 Ќ	142 Ћ	143 Џ
9	1001	144 ђ	145 '	146 '	147 "	148 "	149 ·	150 –	151 —	152	153 ™	154 љ	155 ›	156 њ	157 ќ	158 ћ	159 џ
A	1010	160	161 Ў	162 ў	163 Ј	164 ¤	165 Ґ	166 ¦	167 §	168 Ё	169 ©	170 Є	171 «	172 ¬	173 -	174 ®	175 Ї
B	1011	176 °	177 ±	178 І	179 і	180 ґ	181 µ	182 ¶	183 ·	184 ё	185 №	186 є	187 »	188 ј	189 S	190 s	191 ї
C	1100	192 А	193 Б	194 В	195 Г	196 Д	197 Е	198 Ж	199 З	200 И	201 Й	202 К	203 Л	204 М	205 Н	206 О	207 П
D	1101	208 Р	209 С	210 Т	211 У	212 Ф	213 Х	214 Ц	215 Ч	216 Ш	217 Щ	218 Ъ	219 Ы	220 Ь	221 Э	222 Ю	223 Я
E	1110	224 а	225 б	226 в	227 г	228 д	229 е	230 ж	231 з	232 и	233 й	234 к	235 л	236 м	237 н	238 о	239 п
F	1101	240 р	241 с	242 т	243 у	244 ф	245 х	246 ц	247 ч	248 ш	249 щ	250 ъ	251 ы	252 ь	253 э	254 ю	255 я

Рис. 2.11. Структура кода ANSI

00101011 (в строках указана первая половина восьмибитового кода, а в столбцах — вторая), в 16-ричном виде этот код можно записать как 2B. Но в каждой клеточке указано еще какое-то десятичное число, в данном случае 43. Это не что иное как переведенные в десятичную систему или двоичное число 00101011 или 16-ричное число 2B. Действительно,

$$00101011 = 1 + 2 + 8 + 32 = 43;$$
$$2B = 11 + 2 \cdot 16 = 11 + 32 = 43.$$

Если мы спросим вас, какой комбинацией кодируется заглавная буква, то, посмотрев в таблицу, вы легко определите, что русская буква «Ю» имеет код 11011110, или в 16-ричном виде DE, а в десятичном — 222. Итак, этот вариант ANSI-кодировки применяется в Windows для всех текстовых шрифтов, содержащих русские буквы. При работе с компьютером многие пользователи используют как MS DOS, так и Windows, отличающиеся друг от друга помимо прочего системами кодирования символьной информации. Windows содержит стандартные средства для перехода от одной кодировки к другой, которые часто выполняются автоматически и не доставляют особых забот пользователю. Тем не менее об этом необходимо помнить, чтобы осознанно анализировать те внештатные ситуации, которые могут возникнуть при использовании компьютера.

В современных компьютерных системах возможны различные форматы хранения *звуковой информации*. Наша задача — понять, как именно можно кодировать звук. Рассмотрим этот вопрос подробнее.

Раздел физики, изучающий звуковые явления, называется *акустикой*. Источником звука, а точнее звуковых волн, является колеблющееся тело. Чем больше частота этих колебаний, тем выше звук. Вы не задумывались над тем, кто чаще машет крыльями — муха или комар? Ответ очевиден — комар, так как высота звучания у него выше, нежели у мухи. Музыкальные звуки характеризуются как высотой звучания, так и его длительностью. Кстати, одной из первых систем кодирования звуков явилась нотная запись. На нотном стане располагаются основные *1* и дополнительная *2* линейки (рис. 2.12, *а*). На рис. 2.12, *б* представлено обозначение нот.

Вспомним расположение клавиш на клавиатуре фортепиано (рис. 2.13). Каждой ноте в каждой октаве однозначно соответствует определенная частота звуковых волн. Чем правее мы движемся по клавиатуре, тем выше звук, т.е. больше *частота звучания*. Изменить высоту звука в нотной записи можно знаками «диез» и «бемоль», а отменить их знаком «бекар» (рис. 2.14). Черные клавиши, расположенные между белыми на музыкальной клавиатуре, являются *полутонами*,

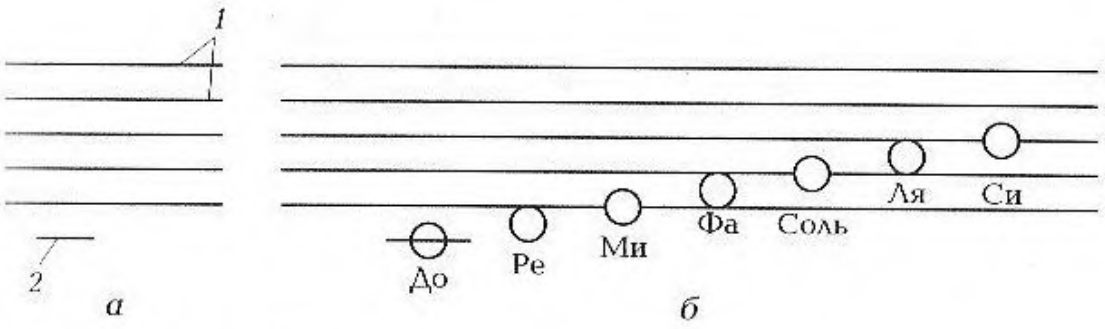


Рис. 2.12. Нотная запись:

а — нотный стан; *б* — звукоряд; *1* — основные линейки; *2* — дополнительная линейка

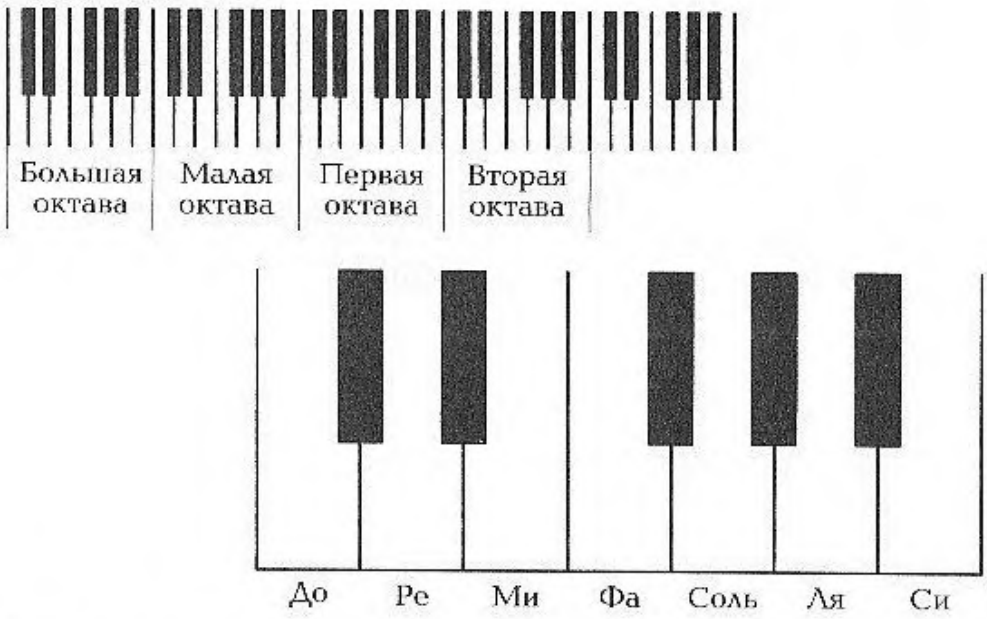


Рис. 2.13. Схема расположения клавиш на клавиатуре фортепиано

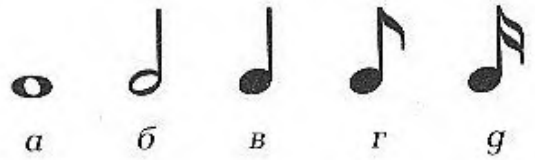
Рис. 2.14. Нотные знаки:

а — диез; *б* — бемоль; *в* — бекар



т. е. при их нажатии воспроизводится звук с частотой, являющейся примерно средним арифметическим частот белых клавиш, между которыми расположена черная. Например, клавиша полутона, расположенная между «до» и «ре», называется «до диез», что означает повышение звука «до» на полтона, или же «ре бемоль», что означает понижение звука «ре» на полтона. Как мы видим, основных нот всего семь, да еще пять полутонов между ними, и все это повторяется. Такая комбинация клавиш и составляет *октаву*.

Теперь о *длительности звучания*. Основная нота — целая. Нота, звучащая в два раза меньше, называется половинной ($^1/_2$), далее идет



а *б* *в* *г* *д*

Рис. 2.15. Обозначение нот:

а — целая; *б* — половинная; *в* — четвертная; *г* — восьмая; *д* — шестнадцатая



а *б* *в* *г* *д*

Рис. 2.16. Обозначение пауз:

а — целая; *б* — половинная; *в* — четвертная; *г* — восьмая; *д* — шестнадцатая

четвертная ($^1/_4$), затем восьмая ($^1/_8$) и т.д. (рис. 2.15). Обозначения соответствующих пауз представлены на рис. 2.16.

Частота звучания задается в герцах (напомним, что человеческое ухо воспринимаст частоты от 16 до 20 000 Гц), а длительность — в миллисекундах.

Частоты для четырех октав представлены в табл. 2.5.

Предположим, что есть фрагмент нотной записи мелодии (рис. 2.17). Используя таблицу, запишем ряд частот, соответствующих каждой

Таблица 2.5

Нота	Октава			
	Большая	Малая	Первая	Вторая
До	130,8	261,63	523,25	1 046,50
Ре	146,83	293,66	587,33	1 174,07
Ми	164,81	329,63	659,26	1 318,05
Фа	174,61	349,63	698,46	1 396,09
Соль	196,0	392,00	784,99	1 568,00
Ля	220	440,00	880,00	1 760,00
Си	247	493,88	987,77	1 975,00
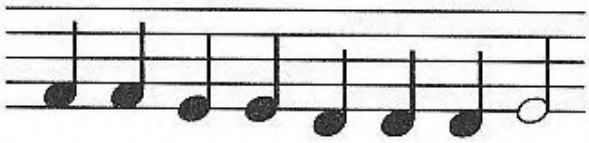


Рис. 2.17. Фрагмент нотной записи

ноте данной мелодии и ряд длительностей звучания каждой из них:

Частота, Гц.....	174,61	174,61	164,81	164,81	146,83	146,83	146,83	164,81
Длительность звучания, мс...	500	500	500	500	500	500	500	1 000

При воспроизведении этого кода прозвучат первые такты народной мелодии «Во саду ли в огороде».

Следует отметить, что чем более скоростным процессором располагает компьютер, тем больше должны быть значения временны́х задержек, поскольку воспроизведение звука кратно скорости компьютера.

Отдельные ноты (звуки) можно объединять (формировать аккорды), добавлять звуковые эффекты, например реверберацию, менять их тембр, представлять в различных форматах. В Windows-приложениях это в основном формат WAV. При звукозаписи осуществляется преобразование акустических волн в тот же формат WAV, широко применяются форматы MP3 и MP4 для записи музыки. Так звук может быть представлен в цифровом коде.

Как известно, *графическая информация* является визуальной. Именно поэтому она находит широкое применение в современном мире. Условно все виды компьютерной графики можно разделить на три класса: растровая, векторная и фрактальная.

Растровая графика чаще всего используется в электронных и полиграфических изданиях, например, практически вся графика в Интернете растровая. Как правило, изображение либо сканируется, либо получается с помощью цифровой фото- или видеокамеры. Поэтому большинство компьютерных программ для обработки графики рассчитаны больше не на создание, а на обработку готового изображения. Основной элемент растровой графики — точка, если она отображена на экране, то это *пиксел*. Пиксел имеет *кодировку цвета* и *координаты*. В зависимости от *разрешающей способности монитора* на экране могут располагаться 640×480, 800×600, 1 024×768 пикселов и т. д. *Разрешение изображения* связано с его размером и определяется количеством точек на 1 дюйм (dots per inch — dpi). Нетрудно посчитать, что для монитора с экраном 15 дюймов в режиме работы 800×600 разрешение составляет 72 dpi. Для печати цветного изображения разрешение должно быть в 3—4 раза выше. Например, стандартный цветной снимок

10×15 см содержит $1\,000 \times 1\,500 = 1\,500\,000$ пикселов. Если на кодирование одной точки затрачивается 3 байт (код цвета и коды двух координат по горизонтали и вертикали изображения), то размер файла, хранящего информацию снимка, составит 4,5 Мбайт.

При использовании растровой графики возникают две основные проблемы:

1) большие объемы графических файлов — это, соответственно, налагает повышенные требования на объемы оперативной памяти и «жестких» дисков и на производительность процессора;

2) эффект пикселизации — заключается в том, что увеличение масштаба просмотра изображения, как правило, приводит к искажению изображения.

Стандартным форматом для хранения изображений растровой графики в Windows является формат BMP (Windows BITMAP — битkарта). Его главный недостаток — отсутствие возможностей сжатия, хотя в последнее время появляются модификации этого формата, обеспечивающие сжатие изображения. Для хранения фотографий используют формат JPEG (Joint Photographic Experts Group — объединенная группа экспертов в области фотографии), который является широко используемым методом сжатия фотоизображений. Самое популярное расширение — .jpg. Оно обеспечивает высокую степень сжатия за счет потери части информации, однако для воспроизведения фотографий на экране этот фактор не является существенным, и, кстати, этим процессом можно управлять. Самый «плотный» из графических форматов, практически не имеющий потерь информации, — GIF (Graphics Interchange Format — формат для обмена изображениями, расширение .gif). Его основное назначение — хранение малоцветных изображений. Для сравнения скажем, что чем меньше цветность изображения, тем менее эффективен формат JPEG и, наоборот, более эффективен GIF. Как мы уже говорили, особые требования к качеству изображения предъявляются в полиграфии. Для этой области создан специальный формат TIFF (Tagged Image File Format — формат хранения растровых графических изображений с высокой глубиной цвета, расширение файлов — .tif). Наиболее распространенным инструментом обработки растровых изображения является программа Adobe Photoshop.

Векторная графика основным элементом имеет не точку, как растровая, а линию, как прямую, так и кривую. В растровой графике *линия* — это совокупность пикселов, и чем линия длиннее, тем больше места в памяти она занимает. В векторной графике в памяти хранятся не точки, а параметры линии, которые не зависят от ее длины. Линия — элементарный объект векторной графики. Все в ней состоит из линий, причем простые объекты объединяются в более сложные, например

треугольник — это три связанных объекта — линии. Именно потому, что основой векторной графики является линия, ее часто называют *объектно-ориентированной*.

Действительно, характеристики линии хранятся в оперативной памяти в виде совокупности параметров, но экран монитора устроен так, что ему в любом случае нужны точки для построения. Каждый раз при выводе линии на экран происходит вычисление координат и цветности этих точек, поэтому векторную графику иначе называют *аналитической* или *вычисляемой*.

Выделяют следующие свойства линии-объекта:

- форма;
- толщина;
- цвет;
- характер (штриховая, сплошная, пунктирная);
- цвет заливки внутренней области (для замкнутых линий);
- текстура (для замкнутых линий);
- карта (для замкнутых линий).

Если линия не является замкнутой, то она имеет две граничные точки, называемые *узлами*, которые в свою очередь имеют параметры. В основе представления линий-объектов в векторной графике лежат математические формулы, например прямой соответствует известная формула $y = kx + b$, кривой второго порядка может соответствовать парабола, гипербола, окружность. Кривым третьего порядка соответствуют уравнения третьего порядка и т.д. Наиболее распространенным редактором, обрабатывающим векторную графику (такие редакторы иногда называют векторными), является CorelDRAW.

Фрактальная графика, как и векторная, является вычисляемой, но информация об объектах в памяти не хранится, а хранятся только математические формулы. Меняя значения коэффициентов в этих формулах, можно получать совершенно другое изображение. При этом новые изображения наследуют свойства родительских, т.е. каждая новая структура повторяет свойства родительской структуры, например снежинка, кристалл, лист папоротника и т.д. На практике средства векторной и фрактальной графики в основном используют для решения конструкторских, проектных и дизайнерских задач. В этих видах графики легко решается *задача масштабирования*, что особенно важно в САПР, при разработке дизайн-проектов и т.д.

При работе с цветом существует очень важное понятие — *цветовое разрешение*. Это метод кодирования цветовой информации. От него, собственно, и зависит количество одновременно отображаемых на экране цветов. Для кодирования черно-белого пиксела достаточно

иметь один бит. Выделение одного байта позволяет кодировать 256 оттенков, два байта (16 бит) дают 65 536 оттенков (режим High Color). И, наконец, три байта (24 бит) дают 16 777 216 оттенков (режим True Color). Как мы знаем из физики, в природе существует семь простых цветов, однако реальные цвета обычно являются комбинацией простых. Такие комбинации принято называть *цветовыми моделями*. В компьютерной технике применяются три цветовые модели, хотя в целом их, конечно, больше.

Цветовая модель RGB (red, green, blue — красный, зеленый, синий) — это основная модель, которая используется в компьютерных мониторах и экранах цветных телевизоров. Считается, что любой цвет состоит из комбинации основных цветов — красного, зеленого и синего. Понятно, что при наложении одного цвета на другой суммарная яркость увеличивается. Сложение трех основных цветов при достаточной яркости дает белый цвет. Новый оттенок получается суммированием основных цветов различной яркости — такой метод называется *аддитивным* (*суммирующим*). Его применяют там, где изображение рассматривается «на просвет»: в мониторах, проекторах, слайдах и т. д. Если интенсивность основного цвета кодировать от 0 до 255, то, очевидно, что при значениях интенсивностей основных цветов: красный — 0, зеленый — 0, синий — 0 экран будет иметь черный цвет (т. е. монитор не будет светиться), а при комбинации интенсивностей: красный — 255, зеленый — 255, синий — 255 цвет будет белым. Понятно, что каждому цветовому оттенку однозначно соответствует определенная комбинация интенсивностей основных цветов.

Цветовая модель CMYK (cyan, magenta, yellow key color — голубой, пурпурный, желтый цветовой ключ) используется для формирования печатных изображений, которые в отличие от экранных смотрятся не в проходящем, а в отраженном свете. Совмещение трех цветов практически полностью поглощает падающий цвет, и изображение получается очень темным. Поэтому здесь используется не сложение цветов, а их вычитание — *субтрактивность*.

Цветовые компоненты в этой модели получаются путем вычитания основных цветов из белого цвета. Полученные цвета называются *дополнительными*, поскольку дополняют основные цвета до белого. Для получения черного цвета (а это одна из серьезных проблем в полиграфии) вводится четвертый компонент — черный цвет.

Цветовая модель HSB (hue, saturation, brightness — оттенок, насыщенность, яркость) наиболее близка человеку (рис. 2.18). Эта модель используется в основном в тех случаях, где изображение создается самим человеком на компьютере. Созданное на компьютере изображение в зависимости от того, для чего оно предназначено,



Рис. 2.18. Цветовая модель HSB

может быть преобразовано в другую модель — для экрана RGB или для печати CMYK, что позволяет делать практически каждый графический редактор. Значение цвета задается вектором, выходящим из центра окружности. Точка в центре соответствует белому цвету, а точки на окружности — чистым цветам. Направление вектора определяет цветовой оттенок и задается в угловых градусах. Длина вектора соответствует насыщенности цвета, а яркость задается отдельной шкалой.

Когда вы пишете графическую программу на каком-либо языке программирования, то с помощью кодов можно задать: цвета текста, фона, линий, палитру (набор цветов), стиль линий (вид, толщину), орнамент заполнения, яркость и многое другое.

Вопросы и задания

1. Как определить, какой вид информации содержится в файле?
2. Сколько байт содержит текст последнего абзаца параграфа?
3. Какие параметры участвуют в кодировании звуковой информации?
4. Как кодируется пиксел на экране монитора?
5. В чем состоит различие и каково назначение форматов RGB, CMYK, HSB? Познакомьтесь с этими форматами подробнее на сайте http://ru.wikipedia.org.

2.12. Файловая система хранения, поиска и обработки информации на диске

У каждого человека есть, условно говоря, три уровня памяти. Во-первых, это быстрая (оперативная) память. Любая обработка информации в мозге человека начинается с ее получения. Быстрая память

не требует запоминания, а лишь поступления информации в нее: например, реакция пешехода на неожиданное препятствие, наша реакция на звонок телефона и пр. Память позволяет на основе поступившей информации принять решение и выполнить действия. Точно так же обстоит дело и с компьютером. Компьютер только тогда может осуществлять процесс обработки информации, когда она находится в его оперативной памяти.

Во-вторых, у каждого из нас есть под рукой записные книжки, органайзеры, мобильные телефоны, которые хранят номера телефонов, адреса знакомых. Эта информация находится у нас «под рукой», и мы в любой момент можем ею воспользоваться. Такие носители информации называются *внешними*. Для компьютера это в первую очередь *диски* (рис. 2.19).

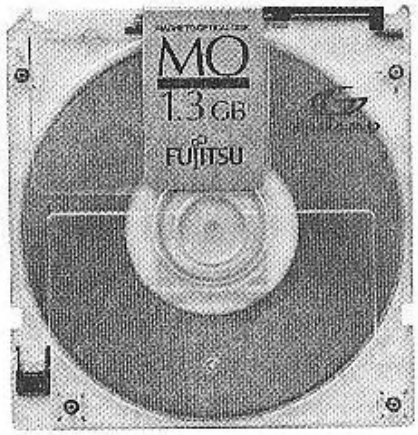
В-третьих, информация, которая находится на внешних носителях, может быть помещена в оперативную память компьютера. На диски может записываться информация из оперативной памяти, которую нужно сохранить после выключения компьютера, так как он при вы-



а



б



в

Рис. 2.19. Носители информации:
а —дискеты и компакт-диски; *б* — «жесткий» диск; *в* — магнитооптический диск

ключении обесточивается и содержимое оперативной памяти обнуляется. Так и человек использует способность мозга и долговременной памяти, которую можно легко восстановить по наводящим словам, образам. Это память, которая требует заучивания (точного запоминания) информации. Например, каждый из нас четко знает и помнит, как его зовут, сколько ему лет, где он проживает и т.д.

Диски бывают «жесткие» (harddisk) магнитные, *несъемные*, т.е. стационарно установленные в системный блок компьютера, и *съемные*, которые могут устанавливаться в нужный для пользователя момент в специальное устройство для работы со съемным носителем информации, например флеш-карты, компакт-диски (CD), магнитооптические диски.

Любая информация в памяти компьютера всегда хранится в виде файлов.

Файл — это поименованный набор данных в памяти компьютера, имеющий признаки начала и конца записи и предназначенный для хранения информации на внешнем носителе.

Для обработки компьютером информация из файла на внешнем носителе должна быть перенесена (*чтение*) в оперативную память компьютера, а также может быть *записана* из оперативной памяти компьютера в файл на внешний носитель. В файлах можно хранить исходные тексты компьютерных программ, программы, готовые к выполнению, различного рода документы, БД, видеоклипы, мелодии, рисунки, фотографии и любые другие данные. Условно файлы подразделяют на текстовые (символьные) и двоичные. *Текстовые файлы* могут быть непосредственно прочитаны человеком. *Двоичные файлы* имеют различные форматы и предназначены для «машинной» обработки. В числе двоичных файлов выделяют *исполняемые файлы* — программы, которые являются готовыми к исполнению в компьютере. При вызове такого файла запускается программа. Примером может служить загрузка компьютера, когда при включении его в сеть начинается автоматический запуск готовых к выполнению программ ОС.

Каждый файл имеет *имя*, которое оформляется строго по правилу: имя файла состоит из двух частей, разделенных точкой, — это *собственно имя* и тип файла — *расширение*. Имя может содержать латинские, русские буквы и цифры. В именах файлов можно использовать практически все символы, кроме так называемых управляющих: /, \, :, *, ?, ", <, > и т.п. Расширение включает в себя до трех символов и указывает на тип данных в файле. Например:

- .exe — исполняемый файл;
- .com — исполняемый файл;

- .bat — командный (пакетный) файл;
- .doc — документ текстового редактора Word;
- .xls — электронная таблица Excel;
- .ppt — презентация PowerPoint;
- .bmp — точечный рисунок;
- .avi — видеофайл;
- .jpg — рисунок;
- .mp3, .wav — звуковые файлы;
- .pas — исходный текст программы на Паскале;
- .c — исходный текст программы на языке Си и т.д.

Таким образом, главным информационным ресурсом компьютерной системы являются файлы на внешних носителях информации. Рассмотрим, как хранятся файлы на *магнитном диске*. Файл — участок дисковой памяти. Это не значит, что этот участок должен быть сплошным. На самом деле чаще он состоит из совокупности отдельных *фрагментов*, потому что в противном случае поверхность магнитного диска использовалась бы неэффективно. Так, при удалении какого-то файла было бы невозможно использовать освободившийся участок, если его размер меньше нового файла, поэтому файлы переносятся на диск фрагментами в имеющиеся свободные участки памяти на диске. Чем меньше фрагментов, из которых состоят файлы, тем меньше временны́х потерь будет при работе компьютера.

Для того чтобы оптимизировать распределение фрагментов памяти диска, периодически необходимо выполнять специальную *программу дефрагментации диска*. Информация о последовательности фрагментов, составляющих каждый файл, хранится в специальной *таблице размещения файлов* FAT (File Allocation Table). Эта таблица размещения файлов находится почти в самом начале любого диска, как «жесткого», так и съемного.

Каждый диск перед началом работы должен быть отформатирован, для чего используются специальные *программы форматирования дисков*. Диск размечен на дорожки — *треки*, в которые записывается информация из оперативной памяти компьютера в соответствии с типом информации и способом ее кодирования.

В процессе форматирования диска в его начало записывается специальная служебная информация, в том числе и две копии таблицы FAT, а также проверяется качество треков и при необходимости производится их отбраковка. Программа форматирования по завершении своей работы выдает результат, где указывает количество пригодных треков и общий объем памяти, доступной на данном носителе информации. Нужно помнить, что форматирование диска практически всег-

да приводит к безвозвратной потере информации, записанной на нем ранее. Обмен информацией между оперативной памятью и внешним носителем осуществляется специальными порциями, называемыми *кластерами*. За один оборот диска считывается или записывается один кластер.

Над файлами можно производить различные операции: их можно создавать и удалять, переименовывать и копировать, пересылать, объединять. Каждый файл перед началом работы с ним должен быть открыт (инициирован), только тогда он становится доступным для работы с конкретной программой обработки информации в файле. После работы файл должен быть закрыт. При этом система обработки информации в файле делает запрос — сохранять ли изменения, полученные в результате обработки.

Размер файла не имеет каких-либо ограничений, он определяется только техническими характеристиками носителя, да еще просто здравым смыслом. Файлам можно *предписывать атрибуты*, т.е. устанавливать некоторые свойства, например «системный», «только для чтения», «архивный», файлы можно делать невидимыми для просмотра, устанавливая атрибут «скрытый». С этими параметрами можно познакомиться, если использовать режим «Свойства» файла.

Поскольку на одном магнитном носителе количество файлов может достигать сотен и более, возникла необходимость упорядочения хранения файлов на дисках. С этой целью были введены *каталоги*. В Windows их принято называть *папками*, а ранее использовался термин «директория». В каталоге группируют файлы по определенной тематике или принадлежности. Например, в одном каталоге хранят документы, в другом — рисунки, в третьем — звуковые файлы с записями любимых мелодий и т.д. Эти папки Windows предлагает пользователю как типовые, например «Мои документы», «Мои рисунки», «Мои видеозаписи» и пр. Количество каталогов также не ограничено. Очень важно помнить, что каталоги на диске имеют *иерархическую структуру*.

На каждом диске обязательно имеется *корневой*, или *главный, каталог*. Он создается в процессе форматирования и может быть только один. Все каталоги, создаваемые в нем, по отношению к нему являются *подкаталогами*. Каждый каталог может хранить как файлы, так и другие каталоги (подкаталоги). На рис. 2.20 в качестве примера показана иерархическая структура хранения информации на диске. В корневом каталоге (папке) имеются три подкаталога «Письма», «Таблицы» и «Музыка», а также записаны антивирусная программа antivirus.exe и русификатор keybru.exe. В каталоге письма имеется подкаталог «Деловые», в котором хранятся три деловых

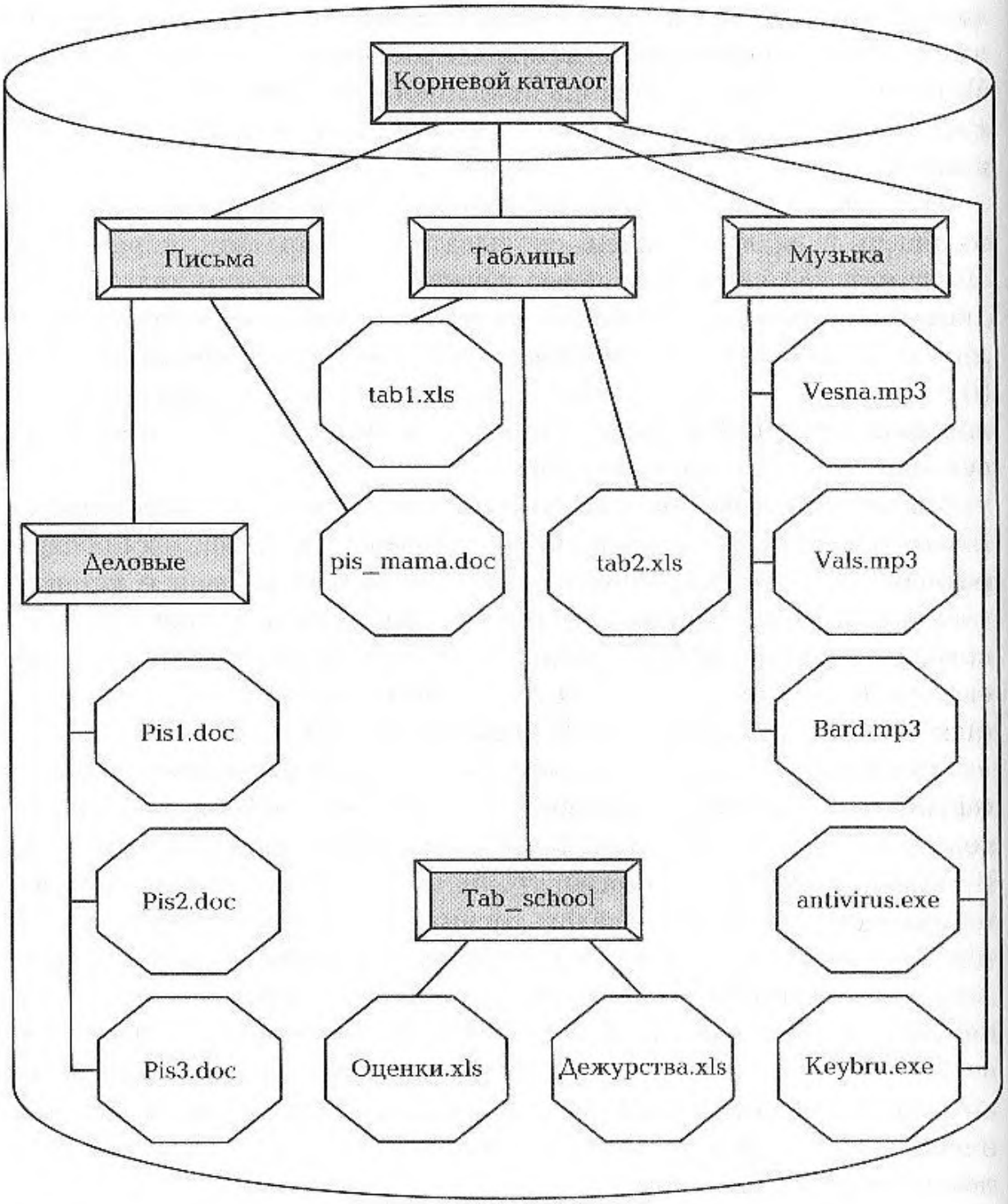


Рис. 2.20. Пример иерархической структуры каталогов на диске

письма, а также непосредственно в нем содержится файл — личное письмо. В каталоге «Музыка» записаны три композиции в формате .MP3.

Таблица FAT содержит информацию о фрагментах, составляющих каждый файл, позволяя работать с файлом, как с единым целым. При работе с MS DOS поначалу использовалась 12-битовая таблица FAT. Она предусматривала работу с накопителями емкостью не более 32 Мбайт. Ее модификация до 16 бит позволила ОС работать с дис-

ками емкостью уже до 128 Мбайт; с внесением в последующем незначительных дополнений этот предел был поднят до 2 Гбайт. Для систем Windows 95 фирма *Microsoft* разработала новое расширение системы FAT — FAT32. Рассмотрим общую структуру файловых систем FAT, а затем остановимся на FAT32.

Эта файловая система предусматривает ряд специальных областей на диске, выделенных для организации пространства диска в процессе его форматирования, — головную запись загрузки, таблицу разбиения диска, запись загрузки, таблицу размещения файлов и корневой каталог. На физическом уровне пространство диска разбивается на 512-байтные области, называемые *секторами*. В системе FAT место для файлов выделяется кластерами, которые состоят из целого числа секторов. Число секторов в кластере должно быть кратно степени числа 2. В *Microsoft* эти кластеры называют *единицами выделения памяти* (allocation unit), а в отчете программы ScanDisk указывается их размер, например 16 384 байт в каждой единице выделения памяти.

Система FAT представляет собой БД, связывающую кластеры дискового пространства с файлами. Первые два элемента содержат информацию о самой FAT. Далее для каждого кластера предусматривается только один элемент, т.е. третий и последующие элементы ставятся в соответствие кластерам дискового пространства, начиная с первого кластера, отведенного для файлов. Элементы FAT могут содержать несколько специальных значений:

- 0000H — кластер свободен, т.е. не использован ни одним файлом;
- FFF7H — кластер содержит один или несколько секторов с физическими дефектами и не должен использоваться;
- FFF8 FFFFH — данный кластер является последним кластером файла.

Для любого используемого файлом, но не последнего кластера элемент FAT содержит номер следующего кластера, занятого файлом.

Каждый каталог, независимо корневой он или подкаталог, также представляет собой БД. В каталоге DOS для каждого файла предусмотрена одна главная запись. В отличие от FAT, где каждый элемент состоит из единственного поля, записи для файла в каталоге состоят из нескольких полей (табл. 2.6). В каталоге 32-байтная запись для файла состоит из восьми полей, содержащих основную информацию о файле, например, его размер, начальный кластер на диске, время и дату последнего внесения изменений. Некоторые поля — имя, расширение, размер, дата и время — можно видеть на экране в списке файлов.

Таблица 2.6

Показатель	Смещение							
	0–7	8–10	11	12–21	22–23	24–25	26–27	28–31
Поле	Имя файла	Расширение	Атрибут	Не используется	Время	Дата	Начальный кластер	Размер файла
Размер, байт	8	3	1	10	2	2	2	4

Когда программа отправляет запрос к ОС с требованием предоставить ей содержимое какого-то файла, ОС просматривает запись каталога для него, чтобы найти первый кластер этого файла. Затем она обращается к элементу FAT для данного кластера, чтобы найти следующий кластер в цепочке. Повторяя этот процесс, пока не обнаружит последний кластер файла, ОС точно определяет, какие кластеры принадлежат данному файлу и в какой последовательности. Таким путем система может предоставить программе любую часть запрашиваемого ею файла. Такой способ организации файла носит название *цепочки FAT*. В системе FAT файлам всегда выделяется целое число кластеров. На «жестком» диске объемом 1,2 Гбайт с кластерами по 32 Кбайт может быть указано при просмотре каталога, что размер текстового файла, содержащего всего два слова «hello, world», составляет всего 12 байт, но на самом деле он занимает 32 Кбайт дискового пространства. Неиспользованная часть кластера называется *потерянным местом* (slack). В небольших файлах почти весь кластер может быть потерянным местом, а в среднем потери составляют половину размера кластера на не используемые, но выделенные файлам области. Один из способов высвобождения пространства на диске — применение программы сжатия диска, например DriveSpace, которая выделяет потерянные места для использования другими файлами.

Рассмотрим, как можно рассчитать размеры секторов и кластеров. В FAT предусмотрены 16-разрядные элементы. Существует 65 536 различных 16-разрядных значений; умножив 65 536 на 512 байт в секторе, получим возможный объем памяти диска — 32 Мбайт. На диске емкостью более 2 Гбайт кластеры имеют размер 64 Кбайт, или 65 536 байт (табл. 2.7). Но самое большое значение, которое может быть представлено в 16 разрядах, составляет на один байт меньше, т.е. 65 535

байт. Таким образом, 64 Кбайт — слишком большая величина для 16 разрядов. В *Microsoft* обнаружили, что при разработке многих существующих в настоящее время программ их авторы исходили из предположения, что значение числа байт в кластере умещается в 16 разрядах. С увеличением размеров диска единица в двоичном восьмиразрядном представлении числа секторов в кластере смещается влево. При смещении единицы за крайнюю левую цифровую позицию возникает ошибка превышения размера 4 Гбайт. Размер кластера указан в виде 16-разрядных двоичных чисел. И вновь единица смещается влево, а при смещении ее за крайнюю левую позицию возникает «2-Гбайт-кризис».

Даже если бы такой проблемы не возникало при объеме в 2 Гбайт, ОС все равно не может работать с дисками емкостью свыше 4 Гбайт. При использовании FAT16 число секторов в кластере сохраняется самой ОС в одном байте блока параметров диска, устанавливаемого ОС. Это число должно быть кратным степени 2 и при этом меньше 256. Таким образом, в кластер может входить не более 128 секторов,

Таблица 2.7

Размер диска	Число секторов в кластере		Размер кластера	
	Десятичная система счисления	Двоичная система счисления	Десятичная система счисления, Кбайт	Двоичная система счисления
Менее 128 Мбайт	4	00000100	2	0000100000000000
Менее 256 Мбайт	8	00001000	4	0001000000000000
Менее 512 Мбайт	16	00010000	8	0010000000000000
Менее 1 Гбайт	32	00100000	16	0100000000000000
Менее 2 Гбайт	64	01000000	32	1000000000000000
Более 2 Гбайт	128	10000000	64	ОШИБКА: Проблема для существующих программ

Таблица 2.8

Емкость диска, Гбайт	Размер кластера, Кбайт
Менее 8	4
Менее 16	8
Менее 32	16
32 и более	32

или 64 Кбайт. В итоге, даже если ограничение в 2 Гбайт снимается, система FAT16 приемлема лишь для дисков емкостью менее 4 Гбайт.

В файловой системе FAT32 как элементы FAT, так и номера секторов — 32-разрядные. Если умножить 4 294 967 296 различных 32-разрядных значений на 512 байт в секторе, то получится огромное число — 2 Тбайт (2 199 023 255 552 байт), которое представляет собой максимально возможную емкость диска при использовании FAT32.

Теперь, когда допустимое число кластеров и число секторов одинаковы, видимо, можно было бы предположить, что для минимизации потерь места на диске в FAT32 будут использоваться кластеры, состоящие только из одного сектора. Но в этом вопросе необходим какой-то компромисс, поэтому в спецификации FAT32 размер кластера для накопителей емкостью до 8 Гбайт составляет 4 Кбайт. В табл. 2.8 отображена зависимость размера кластера от емкости диска.

Одним из факторов, сыгравших роль в принятии такого решения, стало то, что теперь нужно учитывать и размер самой FAT. Операционная система сохраняет две копии FAT, поэтому под элемент каждого кластера в FAT требуется 8 байт. На диске объемом 2 Гбайт FAT займет 32 Мбайт его пространства при размере кластера 512 байт. А если размер кластера составляет 4 Кбайт, для хранения двух таблиц FAT потребуется всего 4 Мбайт, т. е. будет сэкономлено 28 Мбайт.

Чтобы обеспечить возможность работы с возросшим числом кластеров, в записи каталога для каждого файла должно выделяться 4 байт для начального кластера файла (вместо 2 байт в системе FAT16). Традиционно каждая запись в каталоге состоит из 32 байт. В середине этой записи 10 байт не используются (байты с 12 по 21), которые Microsoft зарезервировала для своих собственных нужд в будущем. Два из них теперь отводятся как дополнительные байты, необходимые для указания начального кластера в системе FAT32. Операционная система всегда предусматривала наличие на диске двух экземпляров FAT, но использовался только один из них. С переходом к FAT32

операционная система может работать с любой из этих копий. Еще одно изменение состоит в том, что корневой каталог, раньше имевший фиксированный размер и строго определенное место на диске, теперь можно свободно наращивать по мере необходимости подобно подкаталогу. В настоящее время не существует ограничений на число записей в корневом каталоге. Это особенно важно, поскольку под каждое длинное имя файла используется несколько записей каталога. Сочетание перемещаемого корневого каталога и возможности использования обеих копий FAT — неплохие предпосылки для беспрепятственного динамического изменения размеров разделов диска, например уменьшения раздела в целях высвобождения места для другой ОС. Следует помнить, что поскольку поле для размера файла в записи каталога занимает 32 разряда, размер отдельного файла не может превышать 4 Гбайт.

Мы рассмотрели файловую систему, имеющую название FAT, однако в последнее время все большую популярность приобретает файловая система NTFS (New Technology File System — файловая система операционных систем семейства Windows NT), которая, в частности, используется в Windows NT и Windows XP.

Максимальный размер раздела NTFS в данный момент ограничен лишь размерами «жестких» дисков. Как и любая другая система, NTFS делит все полезное место на кластеры — блоки данных, используемые единовременно. NTFS поддерживает почти любые размеры кластеров — от 512 байт до 64 Кбайт, неким стандартом же считается кластер размером 4 Кбайт. Диск NTFS условно делится на две части. Первые 12 % диска отводятся под так называемую зону MFT (Master File Table). Это БД, представляющая собой общую файловую таблицу, строки которой соответствуют файлам тома NTFS, а столбцы — атрибутам файлов. Запись каких-либо других данных в эту область невозможна. Остальные 88 % диска является обычным пространством для хранения файлов.

Зона MFT поделена на записи фиксированного размера (обычно 1 Кбайт), и каждая запись соответствует какому-либо файлу (в общем смысле этого слова). Первые 16 файлов носят служебный характер и недоступны ОС — они называются *метафайлами*, причем самый первый метафайл — сам MFT. Эти первые 16 элементов MFT — единственная часть диска, имеющая фиксированное положение. Интересно, что вторая копия первых трех записей для надежности — они очень важны — хранится ровно посередине диска. Остальной MFT-файл может располагаться, как и любой другой файл, в произвольных местах диска — восстановить его положение можно с помощью его самого, «зацепившись» за самую основу — первый элемент MFT.

Первые 16 файлов NTFS носят служебный характер. Каждый из них отвечает за какой-либо аспект работы системы. Преимущество настолько модульного подхода заключается в поразительной гибкости. Так, на FAT-диске физическое повреждение в самой области FAT фатально для функционирования всего диска, а на NTFS-диске можно обойти любые неисправности поверхности, кроме первых 16 элементов MFT.

Метафайлы находятся в корневом каталоге NTFS-диска — они начинаются с символа имени «$». В табл. 2.9 приведены используемые в данный момент метафайлы и их назначение.

Таблица 2.9

Метафайл	Назначение
$MFT	Сам MFT
$MFTmirr	Копия первых 16 записей MFT, размещенная по середине диска
$LogFile	Файл поддержки журналирования
$Volume	Служебная информация — метка тома, версия файловой системы и т.д.
$AttrDef	Список стандартных атрибутов файлов на томе
$	Корневой каталог
$Bitmap	Карта свободного места тома
$Boot	Загрузочный сектор (если раздел загрузочный)
$Quota	Файл, в котором записаны права пользователей на использование дискового пространства (начал работать лишь в NT5)
$Upcase	Файл — таблица соответствия прописных и срочных букв в имени файлов на текущем томе. Нужен в основном потому, что в NTFS имена файлов записываются в Unicode, что составляет 65 000 различных символов

Каждый файл на NTFS в общем-то имеет несколько абстрактное строение — у него нет как таковых данных, а есть *потоки* (streams). Один из потоков носит привычный для нее смысл — это данные файла. Но большинство атрибутов файла — тоже потоки! Таким образом, получается, что базовая сущность у файла только одна — номер в MFT. Данная абстракция может использоваться для создания довольно удобных вещей, например, файлу можно «прилепить» еще один поток, записав в него любые данные (информацию об авторе и содержании файла, как это сделано в Windows). Ну и напоследок: имя файла может содержать любые символы, включая полный набор национальных алфавитов. Максимальная длина имени файла 255 символов.

Каталог в NTFS представляет собой специфический файл, хранящий ссылки на другие файлы и каталоги, создавая иерархическое строение данных на диске. Файл каталога поделен на блоки, каждый из которых содержит имя файла, базовые атрибуты и ссылку на элемент MFT, который уже предоставляет полную информацию об элементе каталога. Внутренняя структура каталога представляет собой бинарное дерево. Бинарное же дерево располагает имена файлов таким образом, чтобы поиск файла осуществлялся с помощью получения двухзначных ответов на вопросы о положении файла. На вопрос: в какой группе относительно данного элемента находится искомое имя, бинарное дерево дает ответ: выше или ниже. Вот что это означает: для поиска файла с данным именем в каталоге на FAT-диске ОС прихо-

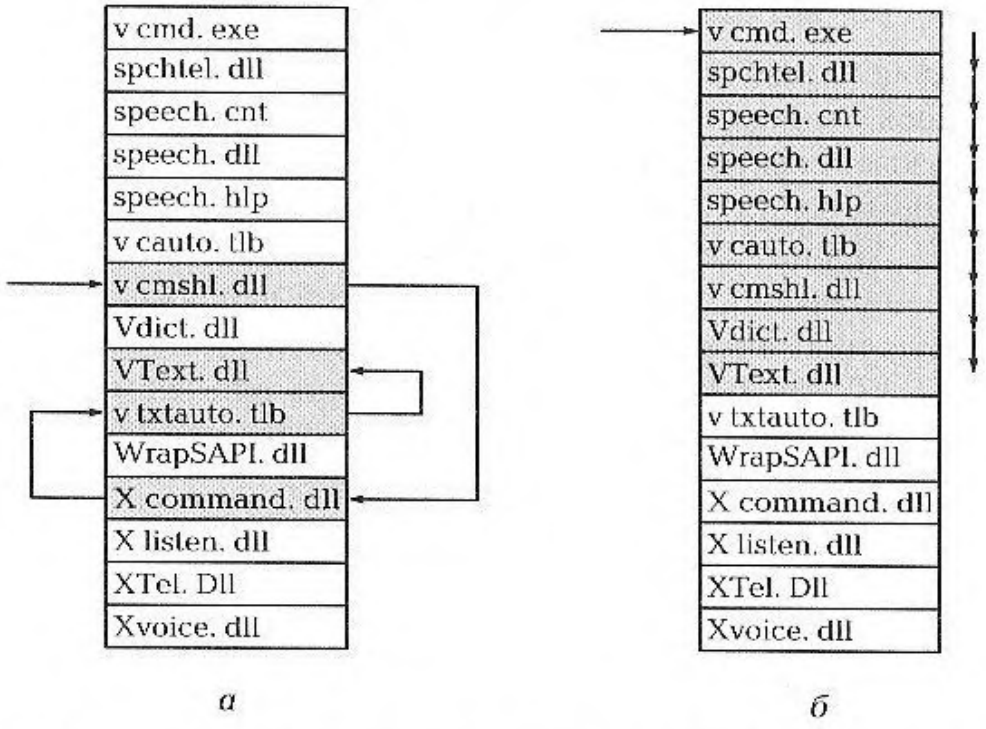


Рис. 2.21. Схемы поиска файла в каталоге: *а* — на NTFS-диске (поиск в дереве); *б* — на FAT-диске (поиск перебором)

дится просматривать все элементы каталога, пока она не найдет нужный. На NTFS-диске поиск начинается со среднего элемента, и каждый ответ сужает зону поиска в среднем в два раза (рис. 2.21).

Для поиска одного файла среди 1 000 файлов на FAT-диске придется осуществить в среднем 500 сравнений (наиболее вероятно, что файл будет найден на середине поиска). На NTFS-диске поиск на основе бинарного дерева будет закончен через 10 шагов (так как 1 024 — это 2^{10}). Экономия времени поиска налицо.

Файлы NTFS-диска имеют один довольно полезный атрибут — «сжатый». Дело в том, что у NTFS есть встроенная поддержка сжатия дисков — то, для чего раньше приходилось использовать специальные программы (Stacker или DoubleSpace).

Рассмотрим достаточно интересный и важный момент — фрагментацию и дефрагментацию NTFS. Как известно, система сильнее всего фрагментирует файлы, когда свободное место кончается и приходится использовать мелкие «дырки», оставшиеся от других файлов. Алгоритм действий при любой записи такой: берется какой-то определенный объем диска и заполняется файлом до конца. Причем в очень интересной последовательности: сначала заполняются большие дырки, потом маленькие, т. е. типичное распределение фрагментов файла по размеру на фрагментированной NTFS выглядит следующим образом (размеры фрагментов):

16 — 16 — 16 — 16 — 16 — [скачок назад] — 15 — 15 — 15 — [назад] — 14 — 14 — 14 ... 1 — 1 — 1 — 1 — 1...

Так процесс идет до самых мелких дырок в один кластер.

Хотелось бы сказать, что если ваша ОС — Windows 2000 и выше, то использовать какую-либо файловую систему, отличную от NTFS, — значит существенно ограничивать свое удобство и гибкость работы самой ОС. Windows составляет с NTFS как бы две части единого целого — множество полезных возможностей напрямую завязано на физическую и логическую структуру файловой системы, и использовать там FAT или FAT32 имеет смысл лишь для совместимости — если у вас стоит задача читать эти диски из каких-либо других систем.

Вопросы и задания

1. Что такое файл? Что показывает расширение в имени файла?
2. Что представляет собой файловая система на диске?
3. Что такое каталог?
4. Что такое кластер?
5. Охарактеризуйте системы FAT, FAT32 и NTFS, используя табл. 2.10.

Таблица 2.10

Параметр		
FAT	FAT32	NTFS
Системы, ее поддерживающие		
DOS, Windows 95, NT всех версий	Windows 98, 2000	NT4, NT5, Windows XP
Максимальный размер тома, Гбайт		
2	Практически не ограничен	Практически не ограничен
Максимальное число файлов на томе		
Примерно 65 000	Практически не ограничено	Практически не ограничено
Имя файла		
С поддержкой длинных имен — 255 символов, системный набор символов	С поддержкой длинных имен — 255 символов, системный набор символов	255 символов, любые символы любых алфавитов (65 000 разных начертаний)
Возможные атрибуты файла		
Базовый набор	Базовый набор	Все, что придет в голову производителям программного обеспечения
Безопасность		
Нет	Нет	Да (с NT5.0 имеется возможность физически шифровать данные)
Сжатие		
Нет	Нет	Да

Окончание табл. 2.10

Параметр		
FAT	FAT32	NTFS
Устойчивость к сбоям		
Средняя (система слишком проста и поэтому ломаться особо нечему)	Плохая (средства оптимизации по скорости привели к появлению слабых по надежности мест)	Полная — автоматическое восстановление системы при любых сбоях (не считая физические ошибки записи, когда пишется одно, а на самом деле записывается другое)
Экономичность		
Минимальная (огромные размеры кластеров на больших дисках)	Улучшена за счет уменьшения размеров кластеров	Максимальная. Очень эффективная и разнообразная система хранения данных
Быстродействие		
Высокое для малого числа файлов, но быстро уменьшается с появлением большого числа файлов в каталогах. Результат — для слабо заполненных дисков — максимальное, для заполненных — плохое	Полностью аналогично FAT, но на дисках большого размера (десятки гигабайт) начинаются серьезные проблемы с общей организацией данных	Система не очень эффективна для малых и простых разделов (до 1 Гбайт), но работа с огромными массивами данных и впушительными каталогами организована как нельзя более эффективно и очень сильно превосходит по скорости другие системы

П-Т Проектное задание

Подготовьте инструкцию по проверке и дефрагментации компьютерного диска с учетом имеющейся на компьютере ОС. Включите в инструкцию раздел по форматированию *съемного жесткого диска* пользователя с выбором соответствующей файловой системы.

2.13. Основы алгоритмизации

Алгоритм — ключевое понятие информатики, это то, без чего ее просто не может быть. Русские слова «метод», «способ», «рецепт», «план действий» можно, правда, с некоторой долей условности, считать близкими словами.

Доналд Кнут, автор всемирно известного сборника «Искусство программирования» считает, что слово «алгоритм» произошло от имени автора известного арабского учебника по математике — Мухаммед бен Муса ал-Хорезми (Abu Ja'far Mohammed ibn Musa al-Khowarizmi) (рис. 2.22). В учебнике были сформулированы, по сути, правила четырех арифметических действий в привычном нам сегодня виде. Фамилия автора в русском переводе звучит как аль-Хорезми, и принято считать, что от нее и произошло как слово «алгоритм», так и слово «алгебра».

Приблизительно в 850 г. н.э. арабский ученый-математик Мухаммед бен Муса ал-Хорезм (из города Хорезма на реке Аму-Дарья, родился около 783 г.) написал книгу об общих правилах решения арифметических задач с помощью уравнений. Она называлась «Китаб ал-Джебр» («Правила восстановления и преобразования»). Эта книга дала имя науке алгебре. Очень большую роль сыграла еще одна книга ал-Хорезми, в которой он подробно описал индийскую арифметику. Триста лет спустя (в 1120 г.) эту книгу перевели на латинский язык, и она стала первым учебником «индийской» (т.е. нашей современной) арифметики для всех европейских городов.

Итак, **алгоритм** — это последовательность *команд* (предписаний инструкций) некоторому *исполнителю*, выполнение которых приводит к получению конечного результата (достижению цели).

Рассмотрим пример. Робот-манипулятор (исполнитель) на заводе собирает детские пирамидки из цветных колец, надевая их в заданной последовательности на стержень. Цвета колец перенумерованы от 1 до N. Кольца каждого i-го цвета лежат в емкости с тем же номером. Какова последовательность его действий для набора колец на стержень?

«Начало». Установить в держатель стержень.

Шаг 1. Начать сборку с выбора емкости $i = 1$.

Шаг 2. Взять кольцо из i-й емкости.



Рис. 2.22. Ал-Хорезми

Шаг 3. Надеть его на стержень.

Шаг 4. Определить — все ли кольца надеты?

- Если нет, то шаг 5. Выбрать следующую емкость $i + 1$ и идти к шагу 2.
- Если да, то отпустить стержень с готовой пирамидкой и идти к шагу «Начало».

Попробуем выразить эту последовательность графически. Графическое отображение алгоритма с помощью блок-схемы называется его *структурной схемой.*

Будем использовать фигуры (блоки):



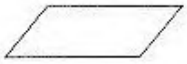
— начало или конец процесса;



— какое-либо действие или процесс;



— проверка условия для принятия решения;



— ввод (вывод) данных.

Тогда структурная схема нашего алгоритма будет выглядеть так, как показано на рис. 2.23.



Рис. 2.23. Структурная схема алгоритма для робота-манипулятора

При этом исполнитель работает автоматически по алгоритму, точно выполняя команды, а от человека зависит правильность размещения цветов на пирамидке, поскольку в этой схеме именно человек наполняет емкости кольцами нужного цвета. Итак, подчеркнем, что алгоритм состоит из шагов. Это очень важное понятие, потому что *шаг* — это элементарное действие, и оно должно быть правильно определено.

Рассмотрим еще один пример — алгоритм, изложенный Евклидом (алгоритм Евклида), описывающий процесс нахождения наибольшего общего делителя двух чисел.

Даны два целых положительных числа m и n. Требуется найти их наибольший общий делитель (НОД), т. е. такое наибольшее целое положительное число, на которое нацело делятся данные числа.

Шаг 1. *Нахождение остатка.* Разделим m на n. Пусть остаток от деления равен r ($0 \le r \le n$).

Шаг 2. *Проверка на ноль.* Это ноль? Если да, то алгоритм заканчивается, и значит n — искомое число. Если нет, продолжаем выполнение.

Шаг 3. *Замена.* Значение n подставим в m, а значение r подставим в n и вернемся к шагу 1.

Алгоритм должен иметь точку входа — «начало» и точку окончания выполнения команд «конец», и так как любой алгоритм пишется для исполнителя, то этому исполнителю нужно сообщить (ввести) значения чисел m и n, позволяющих начать работу исполнителя корректно. На структурных схемах *ввод* или *вывод* информации указывается в фигуре «параллелограмм». Графически, т.е. с помощью структурной схемы, алгоритм Евклида отображен на рис. 2.24.

Как мы можем заметить, в структурной схеме появился еще участок «Отображение r» — это вывод результата исполнения алгоритма в понятной для человека форме. Таким образом, алгоритм Евклида строится по классической схеме: ввод исходных данных — обработка — вывод результатов.

Рассмотрим свойства алгоритмов.



Рис. 2.24. Структурная схема алгоритма Евклида

1. *Конечность.* Алгоритм должен всегда заканчиваться после конечного, пусть даже большого числа шагов. Это очень важное свойство алгоритмов. Именно оно отличает собственно алгоритм от вычислительного (математического) метода.
2. *Определенность или детерминированность.* Каждый шаг алгоритма должен быть точно определен. Действия, которые предписывается произвести исполнителю алгоритма, должны быть строго и однозначно описаны для каждого возможного случая. Определенность алгоритма обусловлена определенностью действий для каждого из его шагов, которые должны быть максимально формализованы в математическом смысле. Для реализации алгоритмов компьютером были созданы специальные *алгоритмические языки*. Записанный на таком языке алгоритм уже является *компьютерной программой*, вернее ее исходным текстом.
3. *Доступность.* У каждого исполнителя, будь то человек, управляющий каким-то механизмом или выполняющий действия на конвейере, техническое устройство (станок с программным управлением, робот-манипулятор, автомат), компьютер, есть своя «система команд», в области которой все понимается и интерпретируется им однозначно. Все, что выходит за эти рамки, даже будучи идеальным алгоритмом по каким-то другим свойствам, не может быть выполнено исполнителем или приведет к серьезным ошибкам.

   Предположим, в научной литературе вы нашли уникальную программу, написанную на языке FORTRAN. Программа вам понравилась, и вы решили использовать ее на своем компьютере. Увы, ваш компьютер не имеет данной *системы программирования*, т.е. исходного языка, компилятора и библиотеки программ. Поэтому, прежде чем использовать ее у себя, вы должны позаботиться, чтобы система команд языка FORTRAN была доступна компьютеру. Следует установить на компьютер соответствующую систему программирования и с ее помощью выполнить на компьютере алгоритм.
4. *Массовость.* Алгоритмы должны быть широко применимыми. Вы знаете, что алгоритм решения квадратного уравнения создал великий французский математик Франсуа Виетт. Зная формулу и подставив в нее соответствующие коэффициенты, можно легко получить значения корней. Таким образом, любой человек, применяя этот алгоритм, может решать квадратное уравнение.
5. *Структура данных.* Алгоритм всегда имеет входные и выходные данные. Способ их описания представляет собой структуру данных. Это могут быть цепочки, списки, деревья, таблицы. Важной

составной частью системы команд при описании алгоритма являются алгоритмические возможности описания структур данных.

6. *Дискретность.* Это означает, что все операции, предусмотренные алгоритмом, должны быть представлены простыми шагами, что каждый шаг можно было выполнить однозначно и за конечный отрезок времени. Очевидно, что для решения одной и той же задачи можно разработать алгоритмы, составленные из различных шагов, однако более эффективным будет тот алгоритм, который приводит к цели быстрее и требует при этом меньше ресурсов исполнителя.

Принято различать несколько типовых алгоритмических конструкций, которые являются как бы кирпичиками для формирования из них сложных алгоритмов: это линейная, разветвляющаяся и циклическая.

Линейная конструкция предполагает процесс выполнения последовательных шагов, причем данная последовательность не изменяется. Это, например, расчет формулы по действиям или последовательный ввод данных для начала работы алгоритма. На структурных схемах линейная конструкция обозначается в виде последовательных стрелок от одного шага к другому (рис. 2.25).

Разветвляющаяся конструкция предполагает выбор шага для выполнения на основе проверки истинности какого-либо условия. Можно провести аналогию с железнодорожной стрелкой, в зависимости от положения которой поезд пойдет по одному из двух путей. Если условие имеет значение «истина», то дальнейшее выполнение пойдет по одной ветке, иначе — по другой. На структурных схемах такие алгоритмы обозначают с помощью ромба с выходом стрелок по итогам проверки условия на истину или ложь (рис. 2.26).

Условием ветвления могут быть сравнение чисел, совпадение набора символов с каким-то заранее заданным словом, получение не-



Рис. 2.25. Обозначение шагов для схемы линейной алгоритмической конструкции

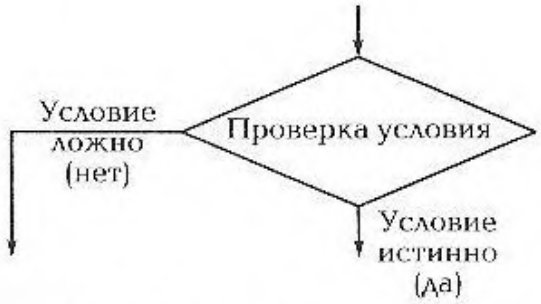


Рис. 2.26. Обозначение шагов для схемы разветвляющейся алгоритмической конструкции

коего информационного признака, например кратности, цвета, веса и т. д. Причем условие может быть представлено в виде логического выражения: *простого*, в частности $a = c$ или $d > 0$, или *составного*, использующего операции алгебры логики. Например, такое: ($a > b$ И $c < d$ ИЛИ $a + c = 0$ ИЛИ $d = 5$). Логическое выражение может принимать лишь одно значение из двух: либо истина, либо ложь. В качестве записи условий в алгоритмической записи часто используют операции отношения: = (равно), <> (не равно), > (больше), < (меньше), >= (больше или равно), <= (меньше или равно), а также логические операции: И (and), ИЛИ (or), НЕ (not), «Исключающее ИЛИ» (xor). Использование составных условий в ряде случаев позволяет повысить эффективность алгоритма за счет его сокращения и упрощения.

Рассмотрим пример: имеются два числа a и b. Если $a > b$, то $c = a + b$, иначе (т.е. если $a < b$ или $a = b$) $c = a\,b$.

Мы уже знаем некоторые свойства алгоритмов и правила их отображения на структурных схемах. С учетом этого получим схему, представленную на рис. 2.27.

Кстати, в известной картине «Витязь на распутье» ее главный герой как раз оценивает условия и принимает решение о направлении своего дальнейшего пути, т.е. реализует алгоритм разветвления.

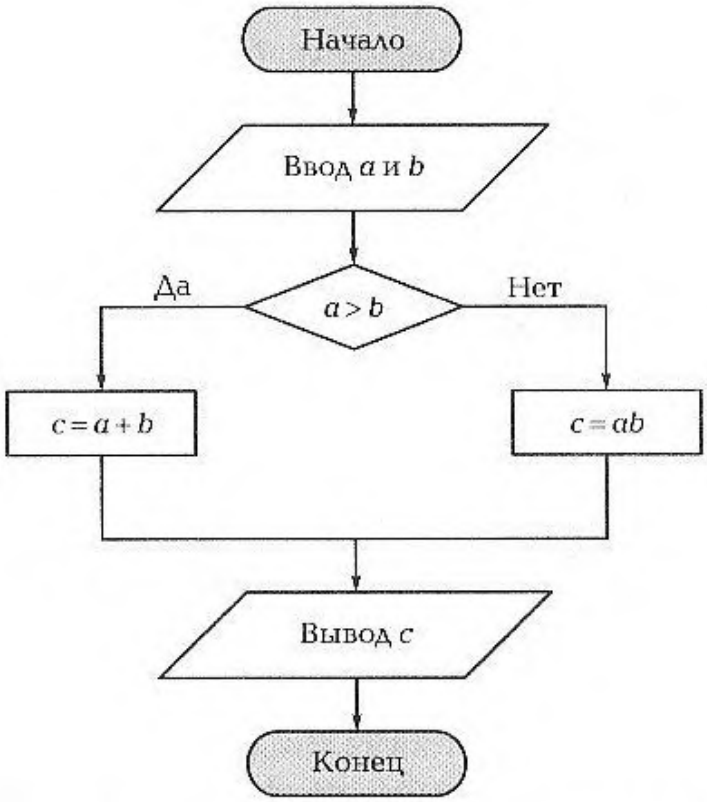


Рис. 2.27. Структурная схема алгоритма, включающего в себя линейную и разветвляющуюся составляющие

Циклическая конструкция описывает некоторый процесс, который многократно повторяется. Эта алгоритмическая конструкция имеет некоторые разновидности: бывают циклы с фиксированным, наперед известным количеством повторений, а бывают с неопределенным. Такие циклы выполняются до тех пор, пока не будет истинно некоторое условие, ограничивающее выполнение цикла.

Рассмотрим цикл с ограниченным числом повторений. Имеется множество разноцветных шаров, 20 из которых необходимо последовательно переложить из одной корзины в другую. На рис. 2.28 представлена структурная схема циклического алгоритма с фиксированным числом повторений. Как мы видим, в этой конструкции используется *счетчик*, значение которого при каждом повторе увеличивается на единицу, и его значение каждый раз сравнивается с числом 20. Когда значение счетчика достигает числа 20 (т.е. переложен 20-й шар), выполнение алгоритма заканчивается.

Таким образом, если цикл задан на конкретное число повторений, то обязательным условием такого цикла является наличие счетчика, т.е. элемента, который ограничивает выполнение алгоритма при достижении нужного значения счетчика.

Рассмотрим цикл с не определенным заранее числом повторений. Допустим, некоторый автомат вырабатывает целые положительные



Рис. 2.28. Структурная схема циклического алгоритма с фиксированным числом повторений

числа в диапазоне от 1 до 1 000 000 (такие автоматы, правда, программные, называют генераторами (датчиками) случайных чисел, хотя, конечно, с математической точки зрения это псевдослучайные числа). Такая операция называется *рандомизацией*. Допустим, что этот процесс должен продолжаться до тех пор, пока не выпадет случайное число, превышающее 600 000, после чего процесс генерации случайных чисел должен быть завершен. Заметьте, нам заранее неизвестно, каким по счету выпадет такое число при генерации случайных чисел. Условие завершения цикла — истинность условия, отражающего наложенное нами ограничение на значение числа.

Структурная схема такого алгоритма представлена на рис. 2.29.

Условием повторения данного алгоритма является *ложность* проверяемого условия, как только оно станет истинным (т.е. полученное число *A* превысит 600 000), выполнение алгоритма закончится, т.е. иными словами цикл повторяется до тех пор, пока условие ложно. Тем не менее очевидно, что число повторений цикла предугадать невозможно, можно лишь утверждать, что сам процесс будет конечным.

Теперь попробуем усложнить задачу. Предположим, что полученное число должно быть больше 600 000, но при этом оно должно быть четным и кратным 65. Попробуем составить структурную схему, используя только простые условия. Давайте посмотрим, сколько чисел из 1 000 000 кратно 65 (1 000 000/65 = 15 385, т.е. 1,54 %), сколько чисел больше 600 000 (1 000 000 – 600 000 = 400 000, т.е. 40 %) и сколько чисел четных (очевидно 50 %). Понятно, что последовательность проверки этих условий в значительной степени определяет время исполнения алгоритма, т.е. один из важнейших показателей эффективности.



Рис. 2.29. Структурная схема циклического алгоритма с неопределенным числом повторений

Поэтому в таких алгоритмах при использовании простых условий нужно вначале проверять условия менее вероятные, что позволит существенно сократить временны́е затраты на его выполнение, поскольку сократит количество проверок условий.

Может возникнуть вопрос: а достигнет ли алгоритм результата, если условия проверять в произвольном порядке? Конечно, достигнет, но за большее время, т. е. такой алгоритм будет менее эффективным. При реализации алгоритмов, обеспечивающих обработку огромных информационных объемов, например при решении статистических задач, такой подход является не просто желательным, а обязательным, поскольку при использовании даже самых высокопроизводительных компьютеров временна́я разница, обусловленная именно эффективностью алгоритма, становится весьма ощутимой. На рис. 2.30 представлена структурная схема алгоритма с неопределенным числом повторений. Условие завершения цикла — истинность одновременно трех условий.

А теперь представим структурную схему циклического алгоритма с неопределенным числом повторений, где условие завершения — истинность составного условия (рис. 2.31).

Несмотря на модификацию, данный алгоритм содержит цикл, повторяющийся при ложности и заканчивающийся при истинности за-

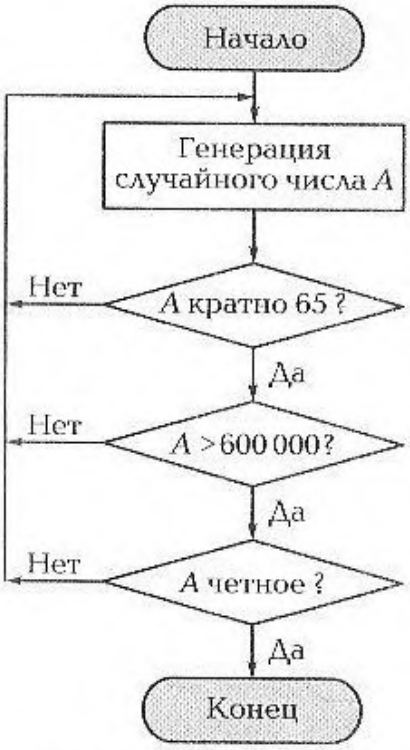


Рис. 2.30. Структурная схема циклического алгоритма: условие завершения — истинность трех условий



Рис. 2.31. Структурная схема циклического алгоритма: условие завершения — истинность составного условия

данного условия. Может быть и обратный подход — продолжение при истинности и завершение при ложности условия, иными словами цикл выполняется до тех пор, пока условие истинно, а завершается, когда условие становится ложным.

Рассмотрим пример. Алгоритм имеет на входе некоторое целое положительное число A, которое делится нацело на 2. Если полученный результат больше 100, то он в свою очередь снова делится на 2, т.е. процесс продолжается до тех пор, пока частное не станет равно 1, и завершается, предварительно выдав количество произведенных делений. Предположим, что $A > 200$. Обозначим B результат деления, а C — счетчик количества делений. На рис. 2.32 представлена структурная схема циклического алгоритма с неопределенным числом повторений. Завершение — по ложности условия.

Следует особо отметить, что циклы могут вкладываться друг в друга, как матрешки. При этом важно помнить, что они не должны пересекаться. На рис. 2.33 показаны вложенные циклы, границы описания которых не пересекаются, а на рис. 2.34 — недопустимые вложенные циклы.

Вложенные циклы имеют широкое применение на практике. Например, в математике существуют функции от многих аргументов типа $w = f(x, y, z, p, q, s)$. Если заданы диапазоны изменения каждого аргумента и, соответственно, шаг этого изменения, то можно найти все

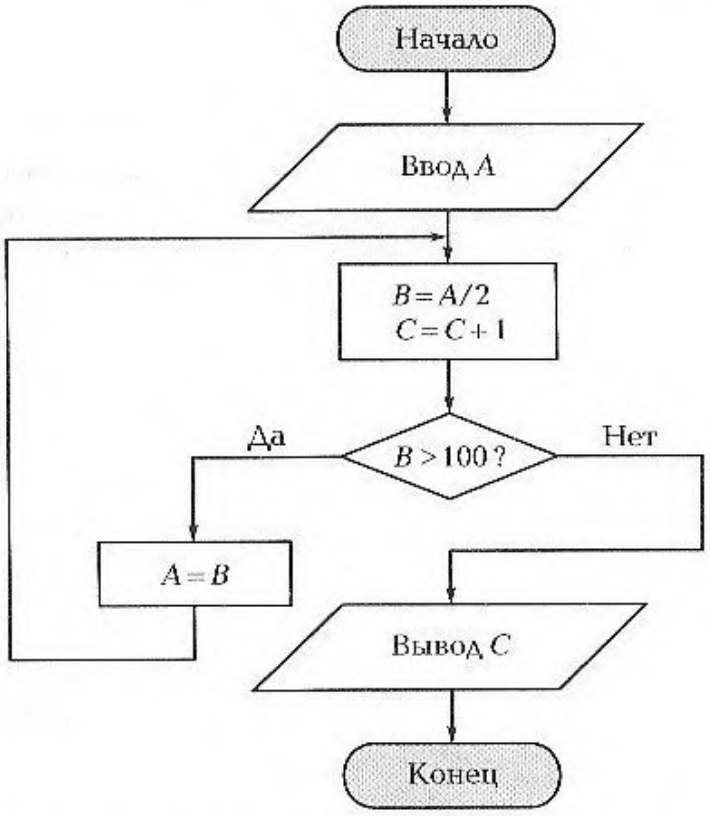


Рис. 2.32. Структурная схема циклического алгоритма: условие завершения — ложность условия

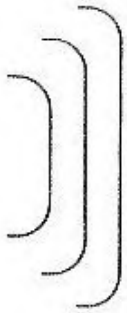


Рис. 2.33. Допустимое расположение областей вложенных циклов



Рис. 2.34. Недопустимое расположение областей вложенных циклов

значения функции. Однако такой алгоритм должен содержать столько циклов, сколько имеется аргументов, поскольку каждый аргумент изменяется только в своем цикле.

Например, задана функция

$$a = \frac{5x^2 + 3y\sqrt{x} + 0{,}26xz}{\sqrt{x + y + x}}.$$

Известно, что

$1 \le x \le 10$ с шагом 1;

$0{,}25 \le y \le 1{,}25$ с шагом 0,5;

$1{,}14 \le z \le 1{,}17$ с шагом 0,01.

Требуется отобразить значения функции для каждой комбинации аргументов и эту комбинацию аргументов. Определим количество шагов циклов по каждому аргументу. Так, по x их будет 10, по y — три (с учетом граничных значений), а по z — четыре шага (также с учетом граничных значений).

Таким образом, заданная функция с учетом шагов имеет $10 \cdot 3 \cdot 4 = 120$ значений. Будем изменять z в самом внешнем цикле, y — в среднем, а x — во внутреннем. При этом очевидно, что сама формула будет вычисляться в самом вложенном цикле.

В каждом цикле для соответствующего аргумента задается его *приращение*, а для вложенных циклов перед входом в них восстанавливается начальное значение аргумента. Структурная схема алгоритма, использующего вложенные циклы, представлена на рис. 2.35.

Как очевидно из примера, хотя число повторений каждого из циклов вполне фиксировано, можно было бы обойтись без явного использования счетчиков каждого цикла, несмотря на то, что в неявном виде он присутствует, поскольку заданы начальные значения аргументов, приращения на каждом шаге их изменения и производится проверка на превышение конечного значения.

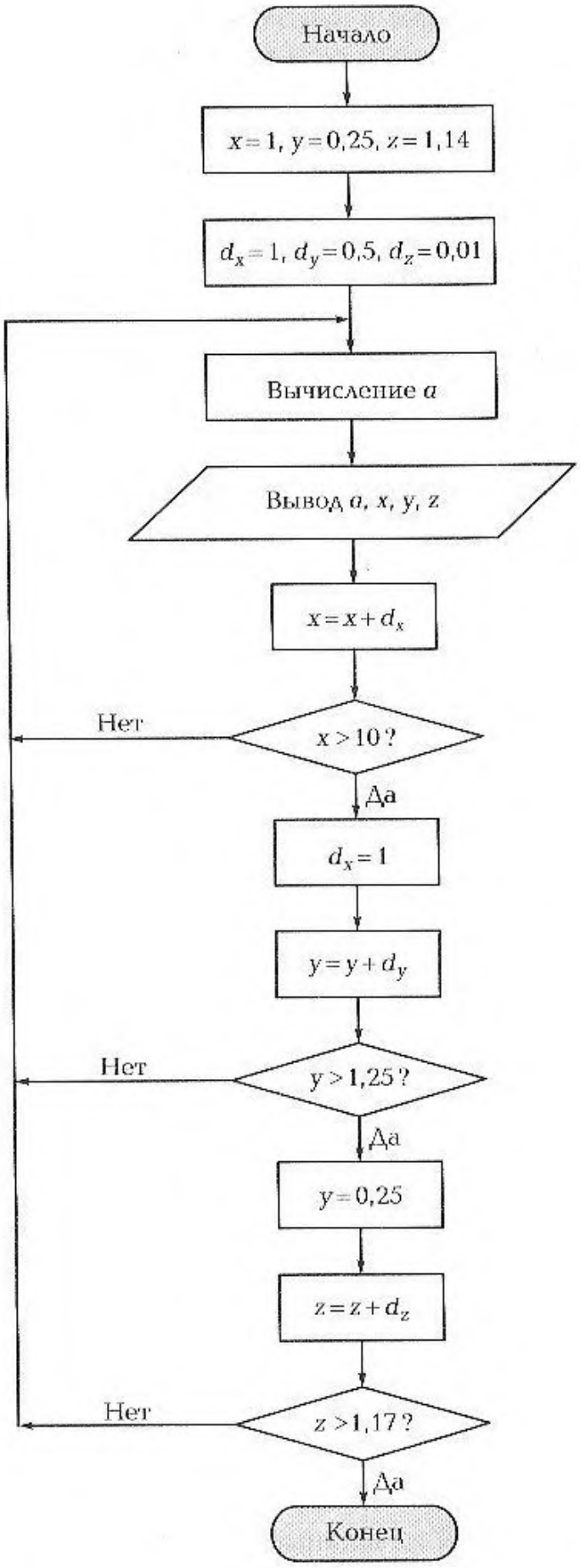


Рис. 2.35. Структурная схема алгоритма, использующего вложенные циклы

Еще один пример. Предположим, что имеется таблица, содержащая 10 строк и восемь столбцов. В этой таблице все числа кратные 5 нужно удвоить, остальные — утроить. Есть два варианта решения. Можно фиксировать строку и перебирать в ней все элементы по столбцам, а можно, наоборот, фиксировать столбец и перебирать в нем все элементы построчно. Заметим, что тот параметр (еще его называют *индексом* — в нашем случае это номера строк и столбцов), который фиксируется, находится в более внешнем цикле. Выберем второй вариант решения. Обозначим номер строки i, а номер столбца j. Структурная схема такого алгоритма будет выглядеть так, как показано на рис. 2.36.

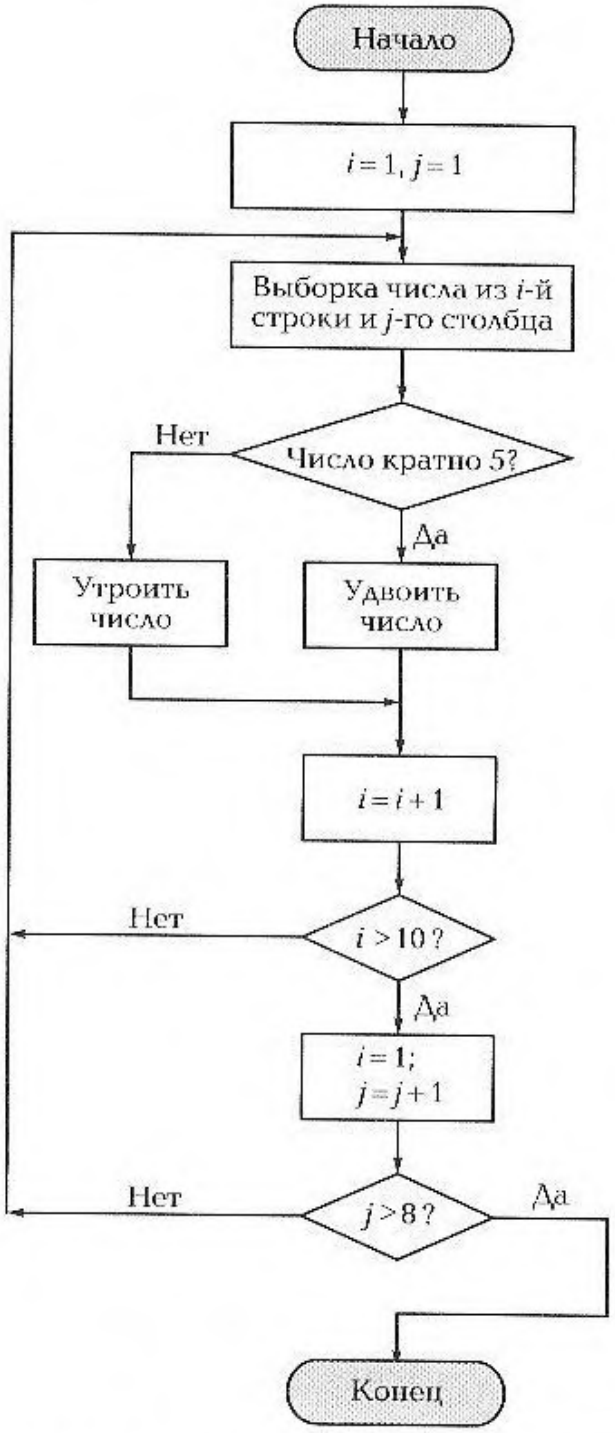


Рис. 2.36. Структурная схема алгоритма, использующего перебор элементов

Вопросы и задания

1. Поясните понятие «алгоритм».
2. В чем состоит особенность описания алгоритмов с помощью структурной схемы и конструкций алгоритмического языка?
3. Перечислите типовые алгоритмические конструкции и объясните их назначение.
4. Что такое исполнитель алгоритма? Что или кто может являться исполнителем алгоритма?
5. Поясните алгоритм работы исполнителя на примере робота-манипулятора или автомата (например, автомата продажи газет).

2.14. Примеры алгоритмов обработки информации

1. *Поиск максимума и минимума в массиве чисел.*

Массив — это упорядоченная последовательность данных, объединенная общим именем, с помощью которого можно обращаться к каждому элементу массива, указывая его номер в списке — индекс. Всем нам знаком классный журнал. В списке класса у каждого из вас есть свой номер. Будем считать, что КЛАСС10А — это название массива, а порядковые номера элементов списка — это индексы. Тогда к ученику Саше Петрову, имеющему порядковый номер 15 в списке, можно обратиться как КЛАСС10А(15), что *однозначно определяет* (*иден-*

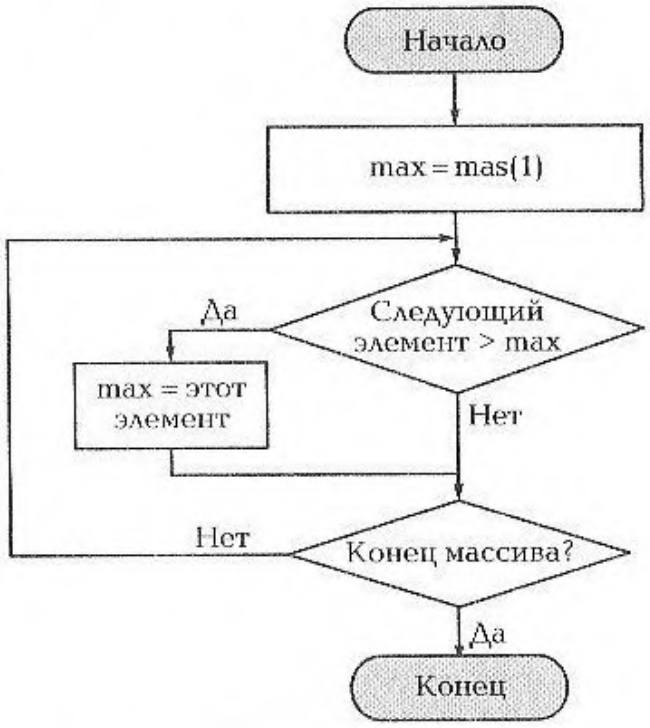


Рис. 2.37. Структурная схема алгоритма поиска максимального элемента массива

шифрует) его в классе, да и в школе. Как найти *минимальный* или *максимальный элемент* массива? Воспользуемся *алгоритмом перебора*, последовательно сравнивая все элементы массива друг с другом, начиная с первого. Пусть массив mas содержит числа — рост учеников в классе. Найдет максимальный рост учеников. Первый элемент mas(1) объявляется максимальным max. Затем каждый элемент, начиная со второго, сравнивается последовательно с этим временны́м максимумом. Если он оказывается больше, то происходит переназначение max. В результате перебора всех элементов массива и их сравнения с max в нем будет находиться истинный максимум массива.

На рис. 2.37 показан алгоритм поиска максимума массива чисел. Очевидно, что для поиска минимума в проверке условия нужно будет использовать знак «меньше» вместо «больше».

2. *Нахождение корней квадратного уравнения.*

Это общеизвестная задача, в которой для уравнения вида

$$ax^2 + bx + c = 0$$

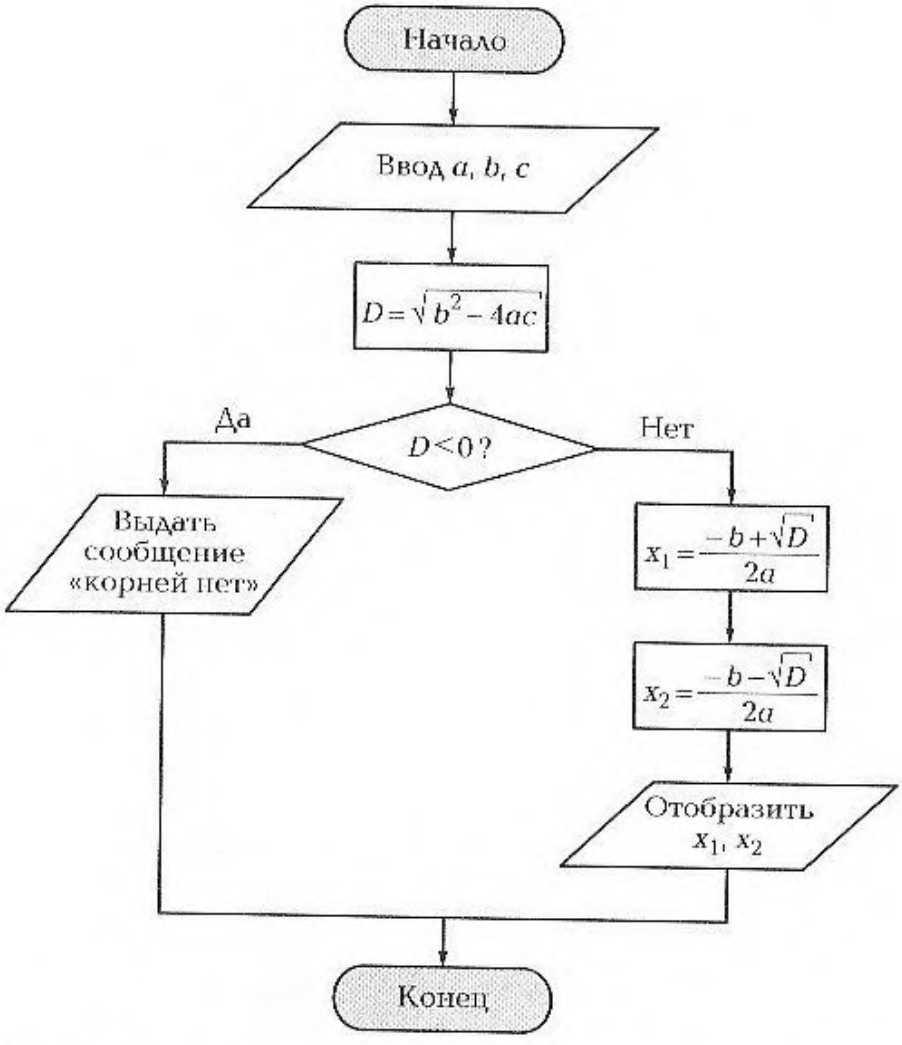


Рис. 2.38. Структурная схема алгоритма поиска корней квадратного уравнения

заданным коэффициентам a, b и c нужно найти корни по формуле при условии, что дискриминант D имеет неотрицательное значение:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Структурная схема этого алгоритма представлена на рис. 2.38.

3. *Упорядочивание (сортировка) элементов массива.*

Требуется расположить элементы массива по *возрастанию* или *убыванию*. Известно несколько алгоритмов сортировки, в данном примере рассмотрим *пузырьковую сортировку*. Суть алгоритма состоит в том, что все элементы массива сравниваются попарно: первый со вторым, второй — с третьим и т. д. Предположим, что необходимо выполнить сортировку массива mas из n элементов по возрастанию

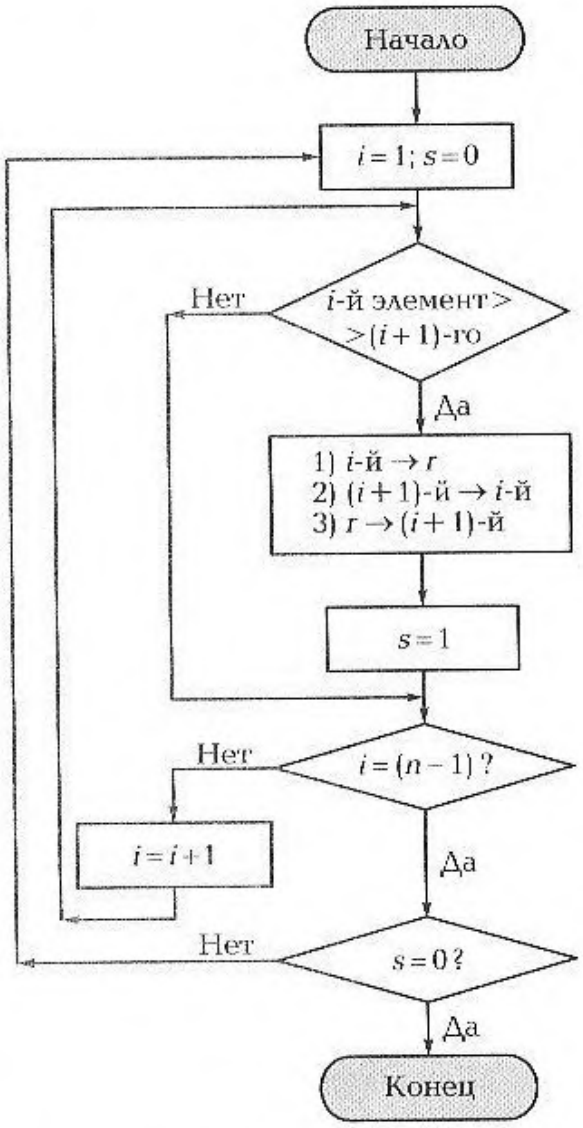


Рис. 2.39. Структурная схема алгоритма пузырьковой сортировки массива по возрастанию

значений элементов. Тогда, если первый элемент больше второго, они меняются местами и в индикатор перестановок s заносится 1, т.е. включается сигнал — перестановка произведена. Этот процесс происходит до тех пор, пока индикатор не будет иметь значение 0, что говорит об отсутствии перестановок, а следовательно, об упорядочивании элементов массива. Вы уже знакомы с понятием i-го элемента массива, значит, следующий по порядку элемент можно обозначить как $(i + 1)$-й. Для того чтобы поменять местами элементы массива, введем еще переменную r для промежуточного хранения значений. Такие переменные иногда называют *рабочими*. На рис. 2.39 представлена схема алгоритма пузырьковой сортировки массива по возрастанию, состоящего из n элементов.

П-СЭ Проектное задание

Постройте структурную схему алгоритма поиска среднего роста учащихся в классе, а также минимального и максимального значений роста. Используйте массив для описания списка учащихся, циклическую алгоритмическую конструкцию для поиска минимума и максимума, суммирования всех элементов этого массива. Результат представьте как итог вычисления среднего арифметического, а для минимального и максимального значений роста учащихся сообщите соответствующие номера этих учащихся в списке класса.

2.15. Системы и технологии программирования

Особое место в ПО компьютера занимают системы программирования. Как правило, система программирования включает в себя три основных компонента:

1) *язык программирования* — язык, на котором пишутся программы;
2) *транслятор* (интерпретатор или компилятор) — программа, «переводящая» язык программирования на язык компьютера;
3) *библиотеку программ* — заранее разработанные программы, реализующие типовые функции и вычислительные методы. В настоящее время таких типовых программ в мире насчитывается несколько тысяч.

Первые системы программирования носили машинно-ориентированный характер. Это были коды, приближенные к работе компьютера. Для обработки компьютером машинных кодов требовались простейшие трансляторы. Однако такие языки программирования достаточно сильно зависели от типа и марки компьютера. Программы представляли собой набор кодов для компьютера, подготовка которых была крайне трудоемким занятием для программистов.

В конце 1950-х гг. было создано несколько алгоритмических языков, самыми известными из которых являлись Algol-60 и FORTRAN-IV. Это были языки программирования, которые позволяли описывать алгоритмы без привязки к конкретной модели ЭВМ и использовали элементы естественного языка — английские слова и символы. Естественно, что значительно усложнились трансляторы, поскольку разница между исходным языком написания программы и машинным языком несоизмеримо возросла.

Первые алгоритмические языки были в основном ориентированы на решение математических, инженерно-технических задач, поскольку именно в этих областях компьютерная техника нашла поначалу самое серьезное применение. Впоследствии появились аналогичные языки для решения экономических задач, задач управления технологическими процессами, проектированием, т.е. они были ориентированы на решение проблем в конкретных областях. Именно это и обусловило их общее название — *проблемно-ориентированные*.

По мере сужения их специализации соответственно росло число языков программирования, сначала оно измерялось десятками, потом — сотнями. Это породило новые проблемы совместимости программ на разных языках программирования при разработке больших систем, и возникла идея создания *универсальных языков программирования*. Одним из таких языков стал PL/1 (Programming Language), который, впрочем, вскоре распался на определенное число подмножеств. Универсальные языки наиболее удобны в использовании, поскольку максимально приближены к естественному человеческому языку, и, конечно, процесс компиляции для них достаточно сложен. Языки этого класса (Basic, Си, Pascal, Ada и др.) объединяет одно — они шаг за шагом последовательно реализуют алгоритм, поэтому их называют *алгоритмическими*, что представляется не совсем корректным, поскольку любое программирование связано с реализацией того или иного алгоритма. Более правильным представляется называть эти языки *процедурно-ориентированными* и также связанную с ними технологию *технологей программирования*.

Широкое внедрение средств компьютерной техники в различные сферы жизни и деятельности человека стимулировало развитие автоматизированных методов и инструментальных средств, применяемых для создания *программного обеспечения*. Производство современного ПО происходит на фоне высоких требований, предъявляемых к качеству создаваемых программ и значительной сложности выполняемых ими функций. «Идеальной» представляется такая технология программирования, которая по некоторому достаточно неформальному описанию объекта программирования автоматически генерирует текст

корректной программы. Разработка любого программного средства может быть представлена как процесс, состоящий из ряда последовательных преобразований одного описания решаемой задачи в другое, начиная от постановки задачи и заканчивая программой, реализованной в кодах конкретного компьютера.

Весь процесс реализации конкретной задачи на компьютере, с определенной степенью условности, можно разделить на 10 этапов:

1. Содержательная постановка задачи.
2. Математическая постановка задачи.
3. Формализация математической постановки — выбор метода решения.
4. Разработка алгоритма решения задачи.
5. Разработка ПО, реализующего алгоритм.
6. Отладка ПО (выявление и устранение ошибок).
7. Тестирование ПО и анализ результатов.
8. Проведение доработок по итогам тестирования.
9. Разработка технической документации (инструкций) для пользователей.
10. Сдача в эксплуатацию.

Все время существования программного средства от зарождения идеи до его создания и эксплуатации обычно определяют как *жизненный цикл*. Укрупненно можно выделить пять наиболее важных этапов жизненного цикла ПО: разработка спецификации (10 %), проектирование (20 %), кодирование (10 %), отладка и опытная эксплуатация (сопровождение) (60 %). В скобках записаны относительные затраты ресурсов на создание ПО.

По затратам времени, человеческих и машинных ресурсов все эти этапы неодинаковы. Наиболее дорогими в этом смысле являются этапы, связанные с *отладкой* — поиском ошибок в программах. Затраты ресурсов на них могут быть равными или даже превосходить совокупные затраты ресурсов на остальных этапах.

Эти затраты быстро увеличиваются при возрастании требований к качеству ПО. На долю устранения ошибок и сопровождение ПО приходится не менее 60 % всех затрат. Следует учитывать, что значительная часть работ, выполняемых на этапе сопровождения, связана с поиском и устранением оставшихся в программе ошибок.

История развития методов и средств автоматизации программирования в этом отношении говорит сама за себя. Первоначально все усилия были сосредоточены на этапе программирования, что позволило решить ряд проблем, связанных с ограничениями функциональных возможностей существующих к тому моменту языков программирования.

На этом этапе следует выделить развитие узкоспециализированных языков программирования, таких как COBOL, Simula, «Симскрипт», «Аспид», «Аналитик» и т.д. Специализированные или проблемно-ориентированные языки, построенные для решения конкретных задач, несомненно, упрощают процесс разработки программных продуктов в выбранной предметной области, но в то же время для новых языков характерна проблема преемственности, а для специализированных — ограниченности области применения.

Постепенное усложнение решаемых задач, а вместе с ними кодов и объемов разрабатываемых программных средств, заставило критически переосмыслить сложившийся к концу 1960-х гг. стиль и технику программирования. В связи с чем, начиная с 1970-х гг., усилиями рабочей группы Технического комитета по программированию Международной федерации по обработке информации, состоящей из таких известных ученых, как Никлаус Вирт, Давид Грис, Эдсчер Дейкстра, У. Дал, Д. Парнас, Чарльз Хоар и руководимой профессором Владиславом Турским, формируются основы методологии и теории программирования. В среде программистов утверждается *принцип структурного программирования* как наиболее производительного стиля (например, язык Си). Развитие концепции структуризации привело к осознанию необходимости структуризации данных, что и определило появление в языках программирования механизмов абстрагирования типов данных (например, «Модула» и др.). Последнее легло в основу *техники модульного программирования*, содержательной базой которой является интерпретация модуля как программной единицы. В модульном программировании акцент делается на разбиение программы на модули таким образом, чтобы данные (обрабатываемые модулем) были спрятаны в нем.

Эволюция техники модульного программирования привела к появлению *объектно-ориентированного стиля программирования*, который во многом унифицировал процесс создания ПО. К достоинствам этого метода относится то, что в нем более полно реализуется технология структурного программирования, облегчается процесс создания сложных иерархических систем, появляется удобная возможность создания пользовательских библиотек объектов в различных областях применения.

Параллельно с развитием процедурного стиля программирования в начале 1970-х гг. появляются непроцедурные языки *логического программирования* и *программирования искусственного интеллекта* (Lisp, Prolog, «Рефал», «Приз»). Программа на таких языках представляет собой совокупность правил, определяющих отношения между объектами и цели (запросы). Сегодня Prolog — язык, предназначенный для

программирования компьютерных приложений, использующих средства и методы искусственного интеллекта, создания экспертных систем и представления знаний.

В последнее время в связи с совершенствованием технических средств отображения информации утверждается новый графический подход к решению проблемы автоматизации разработки ПО, основанный на идее привлечения визуальных форм представления программ, в большей степени соответствующих образному способу мышления человека. Применение графических методов обещает кардинально повысить производительность труда программиста. Кроме того, графическая форма записи по сравнению с текстовым представлением программ обеспечивает более высокий уровень их структуризации, соблюдение технологической культуры программирования, предлагает более надежный стиль программирования.

В настоящее время известно достаточно большое количество удачных инструментальных средств *визуализации программирования*. Прежде всего это относится к визуальным средствам разработки экранных форм, меню и других элементов программы (MS Windows, Visual Basic и т. д.), средствам автоматизации проектирования ПО (CASE-средствам), средствам быстрой разработки приложений для информационных систем (Visual FoxPro), текстовым и графическим редакторам, издательским системам и т. д.

В настоящее время принципиально изменилась роль информационных технологий в обществе. С одной стороны, программные продукты представляют собой достаточно дорогостоящий товар. С другой стороны, ужесточились требования к качеству ПО, поскольку последние широко используются в таких сферах деятельности человека, которые традиционно относят к группе риска. Это аэрокосмические технологии, энергетика, связь, бизнес и т. д. Высокое качество ПО достигается либо методами безошибочного программирования (пассивными методами), либо путем выявления и устранения ошибок (активными методами).

Современные *пассивные методы* безошибочного программирования основываются на реализации процессов автоматизации всех этапов жизненного цикла ПО от проектирования и кодирования программ до документирования и их сопровождения. К таким средствам относятся: CASE-средства, объектно-ориентированное программирование, методы логического программирования, методы визуального программирования, поскольку приближают формы представления программы и способы ее кодирования к образному способу мышления человека.

Активные методы повышения надежности ПО совершенствуются за счет развития средств автоматизации тестирования программ. Раз-

работка систем тестирования программ позволяет максимально сократить число ошибок, допускаемых человеком при разработке программ, и повышает надежность программирования.

? Вопросы и задания

1. В чем состоит особенность процедурно- и проблемно-ориентированных систем программирования?
2. В чем заключается сущность объектно-ориентированной технологии программирования?
3. Каковы основные этапы разработки ПО и что такое жизненный цикл ПО?
4. В чем состоит особенность логического программирования?
5. Каково назначение систем тестирования ПО?

П Проектное задание

Подготовьте рефераt (используя Интернет, выйдите на сайт «Википедия» и задайте в окне поиска фразу «история языков программирования») по истории развития языков программирования «Языки программирования: время, открытия, люди».

2.16. Введение в язык программирования. Синтаксис программы

У каждого языка программирования есть синтаксис и семантика. **Синтаксис** — это совокупность формальных правил написания программ на данном языке, **семантика** — это смысловое значение написанного.

Важным элементом синтаксиса является *алфавит языка*, который представляет собой набор всех допустимых в языке символов.

> Познакомьтесь с синтаксисом языков программирования в разделе «Помощь» (Help) на примере QBasic (рис. 2.40), который размещен по ссылке http://www.softportal.com/get-2408-microsoft-quickbasic-qbasic.html, и Pascal (рис. 2.41), сегодня его наиболее полно представляют версии Turbo Pascal 7.0, Borland Pascal 7.0, Object Pascal, а также Free Pascal (компилятор открыт для использования по ссылке http://www.freepascal.org/ download.var, для работы с ним лучше скачать программу-оболочку, например с сайта http://www.scintilla.org/SciTEDownload.html).

Зарезервированными являются такие слова, смысл которых однозначно трактуется компилятором языка и другим быть не может. Вот

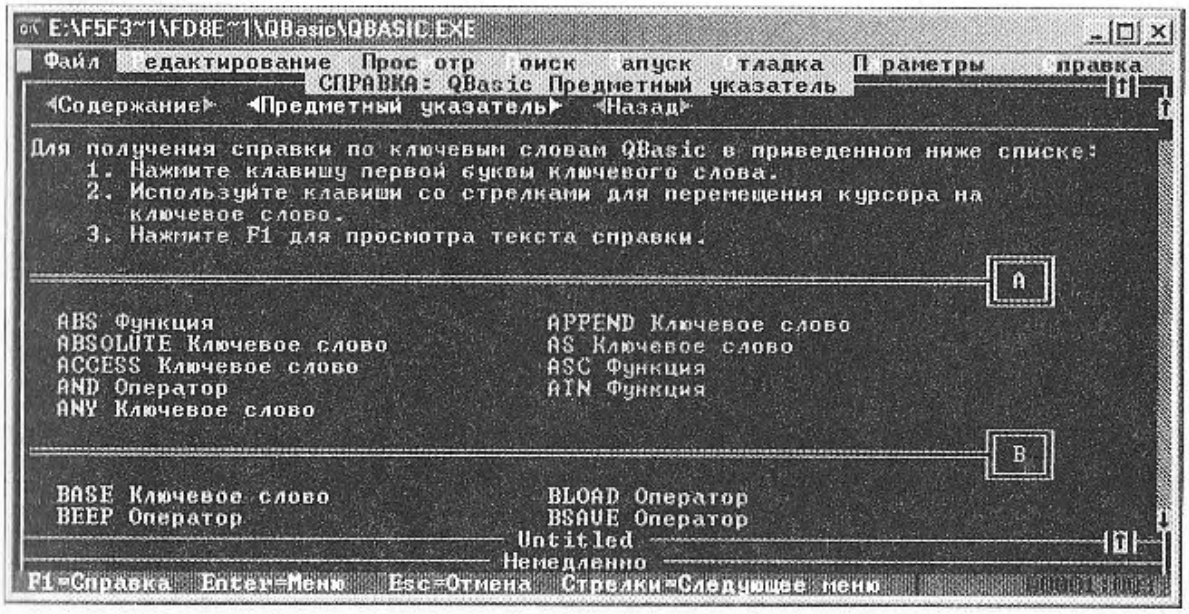


Рис. 2.40. Пример интерфейса услуги «Помощь» в среде QBasic

почему их нельзя использовать иначе, чем по прямому назначению. В отличие от зарезервированных *слова пользователя* задаются самим программистом и им же определяется смысл их использования.

Например, в Turbo Pascal используются следующие зарезервированные слова:

and	*end*	*nil*	*shr*
asm	*file*	*not*	*string*
array	*for*	*object*	*then*
begin	*function*	*of*	*to*
case	*goto*	*or*	*type*
const	*if*	*packed*	*unit*
constructor	*implementation*	*procedure*	*until*
destructor	*in*	*program*	*uses*
div	*inline*	*record*	*var*
do	*interface*	*repeat*	*while*
downto	*label*	*set*	*with*
else	*mod*	*shl*	*xor*

При наборе программы эти слова отображаются на экране белым цветом, напоминая нам, что они являются ключевыми.

Идентификаторы, или имена, могут присваиваться константам, переменным, меткам, типам, объектам, процедурам, функциям, моду-

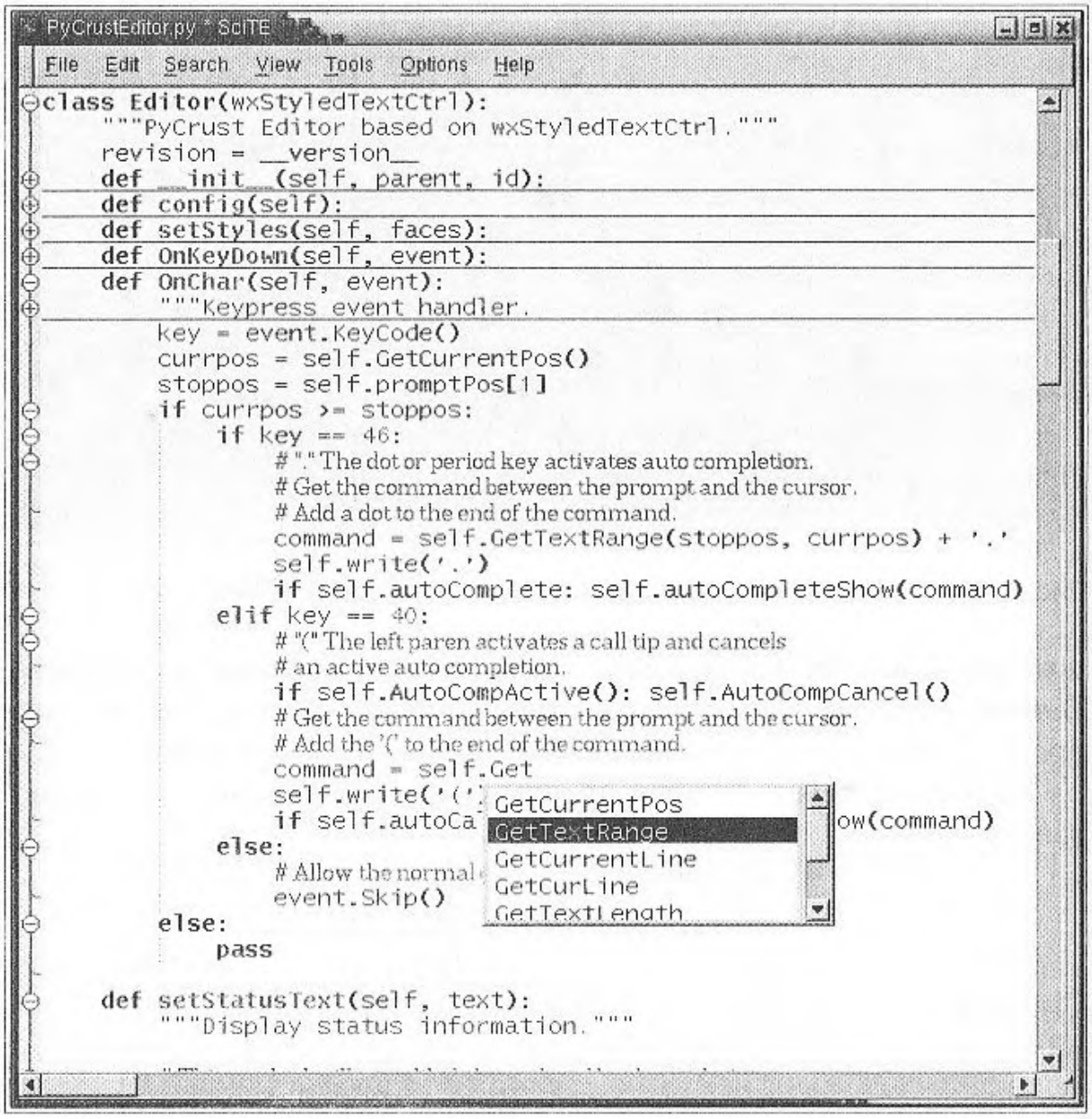


Рис. 2.41. Окно программы-оболочки для Free Pascal

лям, программам, полям записей, иными словами, всему тому, что может быть поименовано. Важно познакомиться с правилами оформления идентификаторов. Любые ошибки в написании имен приведут к синтаксической ошибке, и программа не будет выполняться.

Типы данных — важнейшее понятие языка, поскольку все объекты языка характеризуются типами, которые в значительной степени определяют операции над ними, а также вид «компьютерного» представления соответствующих данных. В языке существует ряд *стандартных типов данных*: целый, вещественный, символьный, логический и пр. Для каждого типа есть правила их обозначения. Эти обозначения пишутся вслед за именем.

Сведения о различных типах данных языка Basic приведены в табл. 2.11.

Таблица 2.11

Тип данных	Обозначение типа в имени	Количество байт в оперативной памяти	Диапазон значений (E = 10, далее указана степень со знаком «+» или «–»)
Целые (integer)	%	2	–32 768...32 767
Длинные целые (long integer)	&	4	–2 147 483 648... +2 147 483 647
Действительные с одинарной точностью (single precision)	–	4	–3.4E+38...+3.4E+38
Действительные с двойной точностью (double-precision)	#	8	–1.8E+308...+1.8E+308
Текстовые (string)	$	1 байт на символ	До 32 767 символов

Правила оформления констант также играют важную роль в синтаксисе языка программирования. Например, при записи десятичных дробей используется десятичная точка, а не запятая, символьные данные оформляются в кавычках, аргумент функции всегда оформляется в скобках, следующих за именем функции, и т. д.

Любой язык программирования имеет целый ряд *встроенных функций*, т. е. готовых программ, одно обращение к которым по их имени приводит к получению результата, например $\sin(x)$, $\cos(x)$, $\log(x)$ и т. д. Напомним, что функция $\mathrm{sqrt}(x)$ обеспечивает нахождение квадратного корня из указанного аргумента. На Turbo Pascal это будет иметь следующий вид:

$$Y:=(sqrt(2.36*x*sqrt(x)+1.23*sqrt(x)+0.25*x+11)/(2.36*sqrt(x))$$

Смысл этой записи (семантика) состоит в вычислении значения выражения, записанного в правой части, в результате которого результат записывается в переменную *Y*.

На примере этой записи мы познакомились с ключевым понятием языка программирования: оператором (командой) языка. Алгоритм, записанный на языке программирования, называется **программой**.

Каждый шаг алгоритма представлен некоторой командой. Команды в программе оформляются по правилам языка программирования и называются *операторами языка программирования*. Заметим, что для любой программы характерен естественный порядок исполнения команд, т.е. команды исполняются в порядке их написания в программе. Этот порядок может быть нарушен командами передачи управления (операторами перехода), которые относятся к управляющим командам, т.е. таким, которые не выполняют непосредственно обработку информации, а управляют работой программы.

Любая программа выполняется в результате ее *трансляции* — перевода записи операторов на язык компьютера. Каждый оператор в программе после трансляции будет представлен набором кодов команд. Эти команды выполняются в оперативной памяти компьютера.

Все константы и переменные размещаются в своих ячейках памяти в соответствии с присвоенными им идентификаторами — *именами*.

Код команды содержит не только эти имена данных, но и адреса ячеек оперативной памяти, в которых размещается значение соответствующих данных, а также код самой операции, предусмотренной в операторе. Операция может быть арифметической (сложение, вычитание, умножение, деление), встроенной функцией (которые сами являют собой набор команд), логической (сравнения) или управления (переход, ввод, вывод, старт, стоп, конец, пауза или задержка).

Интересно отметить еще одну особенность выполнения команд в компьютере. Например, основной арифметической командой является сложение. Операция вычитания представлена в компьютере как сложение с отрицательным числом, а операция умножения — как многократное сложение, соответственно операция деления — как многократное вычитание. Выполняет эти операции в компьютере сумматор. При этом важнейшую роль играет способ представления чисел в компьютере: целых, дробных, положительных и отрицательных. Правила записи этих чисел есть в синтаксисе языка и их следует неукоснительно выполнять.

Операторы языка программирования позволяют приступить к написанию простейших программ с использованием типовых алгоритмических конструкций.

В примере ранее показан оператор присваивания, записанный по правилам языка Turbo Pascal. В этом операторе для обозначения команды «присвоить» используют знак «:=». Математическая запись формулы, записанная по правилам языка программирования справа от знака присваивания, называется *арифметическим выражением*. Арифметические выражения используются повсеместно при работе с компьютером — в программах-калькуляторах, электронных табли-

цах, что будет рассмотрено в дальнейшем. Важную роль в записи арифметического выражения играют встроенные функции, которые сами представляют собой команды для компьютера, требующие вычисления, а также скобки, позволяющие четко определить порядок операций в арифметическом выражении, в том числе для оформления дробного выражения с помощью деления числителя на знаменатель.

Ранее вы познакомились с основными алгоритмическими конструкциями: линейной, разветвляющейся и циклической. Для реализации этих конструкций используются соответствующие операторы языка программирования. Синтаксис (правила) записи операторов в различных языках программирования могут несколько отличаться. В связи с этим можно использовать список правил оформления операторов языка — неотъемлемую часть ПО (системы) языка программирования, представленную в разделе «Помощь». Часто в системе языка программирования автоматически воспроизводится типовой оператор при наборе первых символов оператора, а пользователю требуется его подправить. Рассмотрим основные операторы, реализующие типовые алгоритмические конструкции.

Оператор присваивания. Допустим, есть математическое выражение $Y = \sqrt{6,25x^2 + 2,12x + 0,123}$. Оператор присваивания на языке Basic будет иметь следующий вид:

$$Y=SQR(6.25*x\wedge 2+2.12*x+0.123)$$

Этот оператор работает так: результат вычисления выражения в правой части требуется присвоить в качестве значения переменной *Y*. Именно потому, что данный оператор выполняет функции не только вычисления, но и присваивания, в левой его части не может быть выражения, а только имя одной переменной — ячейки памяти компьютера, в которую производится запись результата вычисления.

> Постарайтесь определить очередность операций в арифметическом выражении, которые вычислит компьютер при выполнении этого оператора. Ясно, что оператор присваивания фактически представляет собой запись линейной алгоритмической конструкции и позволяет выполнить вычисления в соответствии с правилами их очередности, отраженными в записи этого выражения с помощью скобок и имен функций.

Оператор ввода данных. Этот оператор размещает данные в оперативной памяти компьютера. Имена переменных, записанных в операторе *INPUT* в произвольном порядке, получают значения, вводимые

с клавиатуры в этом же порядке, т.е. первой переменной соответствует первое введенное значение, второй — второе и т.д. Синтаксис оператора в общем виде будем называть его *форматом*. Оператор ввода в языке Basic имеет следующий формат:

INPUT <сообщение>; $x_1,x_2,x_3,...,x_n$.

Здесь $x_1,x_2,x_3,...,x_n$ — имена переменных, которые получают значения, вводимые с клавиатуры в порядке их следования. Иначе их называют *списком ввода*; сообщение — это текстовая константа, уточняющая, напоминающая человеку о его действиях.

Оператор вывода. Этот оператор предназначен для вывода результатов или на экран монитора или на принтер. Общий вид (формат) оператора вывода в языке Basic таков:

PRINT $x_1,x_2,x_3,...,x_n$,

где $x_1,x_2,x_3,...,x_n$ — имена переменных, содержимое которых выводится на экран монитора в порядке их следования. Иначе их называют *списком вывода*.

Существуют два вида **операторов (команд) перехода**. *Оператор безусловного перехода* передает управление к другой команде всегда, вне зависимости от каких бы то ни было условий.

Оператор условного перехода передает управление только в случае истинности некоторого условия, а в противном случае — просто игнорируется.

Общий вид (формат) оператора оператора условного перехода в языке Basic таков:

IF <условие> TNEN <оператор или группа операторов>
[*ELSE <оператор или группа операторов>*]

В качестве условия могут быть использованы либо *операции сравнения* (=, <>, >, >=, <, <=), либо *логические выражения*, о которых мы скажем чуть позже. Смысл этого оператора состоит в том, что если условие истинно, то выполняется оператор или группа операторов, следующих за словом *THEN*, а если условие ложно, то выполняется оператор или группа операторов, следующих за словом *ELSE* (иначе). Конструкция *ELSE* здесь заключена в квадратные скобки. По правилам описания форматов это означает ее необязательность. В случае отсутствия в формате конструкции *ELSE* оператор выполняет также действия: если условие истинно, то выполняется оператор или группа операторов, следующих за словом *THEN*, а в противном случае — оператор, следующий за оператором *IF* в программе. Если используется

группа операторов, то они разделяются двоеточиями. Поясним это на примере.

Программа на языке Basic запрашивает и вводит два любых числа, и если первое число больше второго, то выводится их сумма, в противном случае — произведение.

Ниже приведена программа, включающая в себя следующие операторы: ввод первого операнда; ввод второго операнда; условный оператор; вывод результата.

```
INPUT "a="; a
INPUT "b="; b
IF a>b THEN c=a+b ELSE c=a*b
PRINT "c="; c
END
```

Структурная схема алгоритма этой программы была рассмотрена ранее (см. рис. 2.27).

Теперь рассмотрим пример использования так называемой сокращенной формы условного оператора, т.е. не содержащего конструкцию *ELSE*.

Программа запрашивает и вводит с клавиатуры символ. Если это буква «К», то печатать слово «Компьютер»:

```
INPUT "Введите символ"; s$
IF s$="К" THEN PRINT "Компьютер"
END
```

Как мы видим, если вводится другой символ, то программа просто заканчивает работу.

Для реализации циклических алгоритмических конструкций используется **оператор цикла**, в языке Basic это «связка» операторов *FOR* и *NEXT*. Первый из них является начальным и главным оператором. Он открывает собой *тело цикла*, т.е. группу операторов, которые будут циклически выполняться фиксированное число раз.

Формат оператора *FOR* таков:

$FOR\ I = I_{нач}\ TO\ I_{кон}\ [STEP\ \Delta I]$

где I — счетчик цикла; $I_{нач}$ — начальное значение счетчика цикла; $I_{кон}$ — конечное значение счетчика цикла; ΔI — шаг изменения счетчика цикла.

Из формата оператора видно, что конструкция *STEP* является необязательной, если она опущена, то по умолчанию шаг изменения счетчика цикла принимается равным 1. Работа оператора заключается в том, что на каждом шаге к начальному значению счетчика I прибавляется шаг и производится проверка: если $I \le I_{кон}$ для $\Delta I > 0$, $I \ge I_{кон}$

для $\Delta I < 0$, то цикл продолжается, в противном случае выполняются операторы, следующие за телом цикла.

Оператор *NEXT I* является нижним ограничителем тела цикла, которым управляет оператор *FOR* с счетчиком цикла *I*.

Пусть программа запрашивает и вводит с клавиатуры семь целых чисел и выводит на экран сами числа и значения их кубов:

```
FOR i=1 TO 7 STEP 1
INPUT "Введите число"; a
PRINT a, a^3
NEXT i
END
```

Рассмотрим пример обработки массива в цикле. Пусть требуется сформировать одномерный массив из шести случайных целых чисел, значения которых лежат в диапазоне от 0 до 100 (это программный аналог лототрона), вывести элементы массива на экран, а также подсчитать и отобразить на экране сумму всех элементов массива. Ниже приведена программа, включающая в себя следующие операторы: описание массива; описание генератора случайных чисел; оператор цикла с счетчиком *k*; формирование элемента массива; формирование суммы в переменной *s*; вывод сформированного элемента; ограничение области цикла; вывод накопленной суммы.

```
DIM mass(6)
RANDOMIZE TIMER
FOR k=1 TO 6
mass(k)=INT(RND*100)
s=s+mass(k)
PRINT "mass("; k;")="; mass(k)
NEXT k
PRINT "summa="; s
END
```

Возможна организация цикла с неопределенным числом повторений. Рассмотрим пример такой организации в языке Basic средствами оператора *WHILE—WEND*. Формат этого оператора будет иметь следующий вид:

WHILE <условие>
<Тело цикла>
WEND

Принцип работы: до тех пор, пока условие истинно, цикл выполняется, но при наступлении ложности условия программа продолжает выполнение дальше за словом *WEND*.

В предыдущем примере изменим условие: пусть процесс генерации случайных чисел продолжается до тех пор, пока не будет получено число 77. Выведем на экран номер этого числа по порядку поступления от генератора случайных чисел:

```
RANDOMIZE TIMER
WHILE a<>77
a=INT(RND*100)
k=k+1
WEND
PRINT "k="; k
END
```

Вопросы и задания

1. Что включает в себя синтаксис языка программирования?
2. Как классифицируют данные в языке программирования?
3. Что представляет собой программа?
4. Охарактеризуйте оператор присваивания.
5. Охарактеризуйте операторы, реализующие основные алгоритмические конструкции.

Проектное задание

Постройте структурные схемы для приведенных ниже примеров программ. Прокомментируйте каждый оператор. Выполните программы на компьютере в среде QBasic.

1. В числовую переменную последовательно вводятся 10 десятичных цифр. Выведите на экран изображение числа, составленного из введенных цифр, но в порядке, обратном их вводу.

```
t$ = " "
FOR i%=1 TO 10
INPUT "Введите цифру"; a%
s$=STR$(a%)
MID$(t$, 11–i%)=s$
NEXT i%
PRINT t$
END
```

2. Введите текст длиной до 254 символов в виде строки, а затем выведите на экран сообщение о том, является ли этот текст палиндромом или нет, т.е. читается ли он одинаково, без учета пробелов, как слева направо, так и справа налево.

```
INPUT «Введите фразу»; s$
l=LEN(s$)+1
```

```
FOR i=1 TO l
m=l–i
b$=MID$(s$, m, 1)
a$=MID$(s$, i, 1)
IF a$< b$ THEN f=f+1
PRINT MID$(s$, i, 1): PRINT MID$(s$, l–1, 1)
NEXT i
IF f=0 THEN PRINT "Это палиндром" ELSE PRINT "Это не палиндром"
END
```

3. Введите с клавиатуры границы диапазона трехзначных натуральных чисел, из которых нужно отобразить только простые. Определите и отобразите количество таких чисел и их сумму.

```
INPUT "Введите границы диапазона"; x, y
FOR i=x TO y
FOR j=2 TO i–1
IF (i MOD j=0) THEN n=n+1
NEXT j
IF n=0 THEN PRINT i: k=k+1: s=s+i
n=0
NEXT i
PRINT "k="; k, "s="; s
END
```

2.17. Введение в язык программирования. Семантика программы

Программа на языке программирования представляет собой алгоритм, записанный с помощью операторов языка. Собственно алгоритм и является *семантикой программы*. Большую роль играет правильная запись алгоритма средствами синтаксиса языка. Но не менее важную функцию выполняет и правильно построенная структура программы, отражающая порядок выполнения команд алгоритма компьютером.

В каждой программе имеется *описательная* (декларативная) и *исполняемая* часть. Описать имя переменной или функции — в первую очередь присвоить им тип. Покажем это на примере языка Pascal.

Программа всегда начинается с заголовка:

PROGRAMM <имя программы>;

Имя программы выбирается произвольно, но оно должно быть уникальным, т. е. не должно повторять зарезервированные в языке слова и операторы. Далее следует описательная часть, которая может включать в себя несколько разделов.

Раздел описания констант:

CONST {Заголовок раздела описания констант}
F=15; {Целочисленная константа}
A="&"; {Символьная константа}
Lg="Computer"; {Строковая константа}
S12_A=3.8521254; {Вещественная константа}
TT=TRUE; {Булевская (логическая) константа}

Обратите внимание, что комментарии заключаются в фигурные скобки. То же самое можно делать в программе, помещая комментарии везде, где может иметь место пробел. Это часто оказывается полезным для документирования программы, да и просто в качестве памятки. На работу программы они никак не влияют.

Раздел описания переменных начинается с ключевого слова *VAR*, за которым следуют сами описания. Напомним, что переменные — это некоторые имена, которые в процессе работы программы могут принимать различные значения. Например:

VAR {Заголовок раздела описания переменных}
I:integer; {Переменная целая типа INTEGER}
YT:byte; {Переменная целая типа BYTE}
S:real; {Вещественная переменная типа REAL}
B67:char; {Символьная переменная}
STR:string; {Строковая переменная}
R_T:boolean; {Булевская (логическая) переменная}

Обратите внимание, что после имени переменной ставится двоеточие, а любое предложение языка заканчивается точкой с запятой.

Раздел описания типов начинается с ключевого слова *TYPE* и включает в себя описания таких типов данных, которых нет среди стандартных. Например, нам для создания библиотечной картотеки нужно описать такой тип данных, который бы содержал поля *AVT* (автор), *NASV* (название), *GOD* (год издания). Такого типа среди стандартных явно нет, поэтому опишем структуру, которая называется записью (*RECORD*):

TYPE {Заголовок раздела описания типов}
BIBL=RECORD {Называем новый тип BIBL и говорим, что он — запись}
AVT:string; {Поле записи, хранящее фамилию автора, строкового типа}
NASV:string; {Поле записи, хранящее название}
GOD:word; {Поле записи, хранящее год издания}
END; {Любое описание типа заканчивается END}

Если тип описан, то можно смело использовать его наряду со стандартными, например:

```
VAR
SW:integer;
Poi:char;
Z:bibl;
A123:longint;
```

Это означает, что переменная *Z* имеет тип *BIBL*, т.е. состоит из полей *Z.AVT*, *Z.NASV*, *Z.GOD*.

Раздел описания меток начинается с ключевого слова *LABEL*, за которым просто перечисляются метки. Метки ставятся в программе перед теми операторами, на которые осуществляется переход, например, если есть оператор

```
Z:=X+Y;
```

и на него предполагается переход, то его нужно предварить меткой — любым именем, заканчивающимся двоеточием. Тогда оператор будет иметь следующий вид:

```
M11:Z:=X+Y;
```

Эта метка, естественно, должна быть описана в разделе описания меток как:

```
LABEL
M11
```

Если в программе несколько меток, они могут быть описаны в одной строке:

```
LABEL
M11,F15,D,G123,INV
```

Если программа использует самостоятельные программные модули — подпрограммы (процедуры и функции), созданные автором, то они тоже должны быть описаны в описательной части, т.е. до начала исполняемой части:

```
PROCEDURE <имя процедуры> [параметры]
{Описательная часть процедуры}
BEGIN
{Исполняемая часть процедуры}
END;
FUNCTION <имя функции> [параметры]
{Описательная часть функции}
```

BEGIN
{Исполняемая часть функции}
END;

С оператора *BEGIN* (начало) начинается исполняемая часть как процедур и функций, так и основной программы.

В семантике языка Pascal важную роль играет составной оператор — это последовательность любых операторов языка, заключенная в программные скобки, например *BEGIN* и *END*. Составные операторы можно вкладывать друг в друга. При этом важно следить, чтобы их области начала и конца (программные скобки) не пересекались (рис. 2.42).

Рассмотрим примеры программ на языке Pascal.

> В каждом примере постарайтесь выделить описательную и исполнительную части программы и объяснить процесс выполнения программы по ее записи.

Программа «Знакомство» запрашивает имя пользователя, которое вводится с клавиатуры и здоровается с обладателем этого имени. Ввод с клавиатуры будем производить оператором *READLN* (x_1, x_2, x_3, ..., x_n). Вывод информации на экран осуществляется оператором *WRITELN* (x_1, x_2, x_3, ..., x_n). Нужно помнить, что вводимые значения по своему типу должны соответствовать типу переменных оператора:

```
PROGRAM ZNAKOM; {Заголовок программы}
VAR {Заголовок раздела описания переменных}
S:STRING[15]; {Описание строковой переменной длиной 15 байт}
BEGIN {Начало исполняемой части}
WRITELN('Как Вас зовут?'); {Вывод на экран текста}
READLN(S); {Ввод с клавиатуры значения переменной S}
WRITELN(«Здравствуйте, «,S); {Вывод
на экран текста и значения переменной}
END. {Оператор завершения программы}
```

В конце программы после оператора *END* ставится точка.

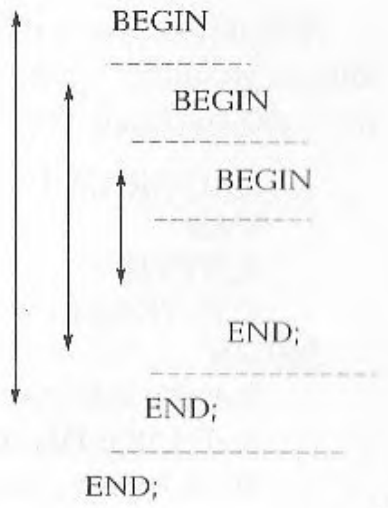


Рис. 2.42. Схема допустимого расположения программных скобок Begin/End

Программа расчета функции $Z = \sin(x+y) + \sqrt{\frac{\ln(x-y)}{e^{\sqrt{x+y}}}} + \operatorname{ctg}(xy)$ запрашивает и вводит с клавиатуры значения аргументов и отображает результат:

```
PROGRAM FUNCXY;
VAR
X,Y:REAL;
BEGIN
WRITELN('Введите значения x и y');
READLN(X,Y);
Z=SIN(X+Y)+SQRT(LN(X-Y)/EXP(SQRT(X+Y)))+
+COS(X*Y)/SIN(X/Y);
WRITELN ('X=',X,'Y=',Y,'Z=',Z);
END.
```

Мы уже упомянули функцию «генератор случайных чисел». Так, если в программе указана функция *RANDOM*(*N*), то будут вырабатываться случайные числа в диапазоне $0 \le x < N$. Для того чтобы эта функция работала правильно, генератор случайных чисел должен быть инициализирован процедурой *RANDOMIZE*. В предложенной ниже программе генератор случайных чисел вырабатывает целые числа в диапазоне от 0 до 100 и выводит их на экран:

```
PROGRAM RAN1;
VAR
A:BYTE;
BEGIN
RANDOMIZE;
A=RANDOM(101);
WRITELN('A=',A);
END.
```

Добавим в программу разветвляющуюся конструкцию, проверяющую условие: если случайное число является четным, то извлечь из него квадратный корень, а если нечетным — возвести в квадрат:

```
PROGRAM RAN2;
VAR
A:BYTE;
C:INTEGER;
BEGIN
RANDOMIZE;
A=RANDOM(101);
IF(A MOD 2 =0) THEN C:=SQRT(A) ELSE C:=SQR(A)
```

```
    WRITELN('C=',C);
END.
```

Рассмотрим циклическую конструкцию программы. Цикл с фиксированным числом повторений реализуется оператором

FOR I:=I_n TO I_k DO <тело цикла>

где I — управляющая переменная цикла; I_n — начальное значение управляющей переменной цикла; I_k — конечное значение управляющей переменной цикла.

Тело цикла обозначено — оператор или группа операторов, выполняется до тех пор, пока $I \le I_k$.

В таком цикле шаг не указывается, поскольку он всегда равен 1.

Существует еще одна модификация этого цикла. Формат оператора:

FOR I:=I_n DOWN TO I_k DO <тело цикла>

В таком цикле шаг не указывается, поскольку он всегда равен –1.

В этих операторах либо тело цикла состоит из одного оператора, либо используются составные операторы — заключенные в программные скобки *BEGIN* и *END*. Операторы цикла также могут быть вложены друг в друга. При этом действует то же правило — области циклов пересекаться не должны.

Пусть, например, генератор случайных чисел вырабатывает 1 000 чисел, лежащих в диапазоне от 0 до 200. Требуется определить и отобразить минимальное и максимальное из выработанных чисел и их порядковые номера. Это позволит сделать следующая программа:

```
    PROGRAM MXMN;
VARMX,
MX,MN,A:BYTE;
I,N1,N2:INTEGER;
BEGIN
MX:=0;
MN:=202;
RANDOMIZE;
FOR I:=1 TO 1000 DO
BEGIN
A:=RANDOM(201);
IF A>MX THEN
      BEGIN
      MX:=A;N1:=I;END
   ELSE
   IF A<MN THEN
```

```
    BEGIN
    MN:=A;N2:=I;END
END;
WRITELN('MAXIMUM=',MX,' ','N=',N1,' ','MINIMUM=',MN,' ',
'N=', N2);
END.
```

Рассмотрим еще один пример цикла с неопределенным числом повторений *WHILE*. Этот цикл имеет формат

WHILE <условие> DO <оператор>

Программа имеет следующий вид:

```
PROGRAM WHILEPRIM;
VAR
I,A:BYTE;
BEGIN
RANDOMIZE;
WHILE(A<>77) DO
BEGIN
A:=RANDOM(101);
INC(I);
END;
WRITELN('I=',I);
END.
```

Сравните запись программы с последним примером из предыдущего параграфа.

Вопросы и задания

1. Как известно, площадь круга рассчитывается по формуле $S = \pi R^2$. Составьте программу нахождения и вывода на экран значения площади круга, если значение радиуса задается с клавиатуры.
2. Составьте программу вывода на экран фразы «XXXXXXXX — выдающийся ученый XX века». Фамилия ученого XXXXXXXX вводится с клавиатуры.
3. Имеется функция $Z = \dfrac{5{,}23x^2 + 4{,}36x + 1{,}2y^3 + 0{,}89y}{2\sin(x+y)e^{xy}}$. Составьте программу нахождения значения функции для значений x и y, вводимых с клавиатуры.
4. Используя структурную схему составьте программу «Пузырьковая сортировка массива».
5. Составьте и выполните программу «Рост и вес среднестатистического учащегося».

П-Т Проектное задание

С помощью блок-схемы опишите семантику представленной далее программы вывода на экран всех значений таблицы кодов ASCII, воспользуйтесь пройденным материалом. В данном примере вы видите два цикла со счетчиками цикла *i* и *j*, причем второй вложен в первый. Проверьте правило вложенности циклов. Выполните программу на компьютере в среде Turbo Pascal, предварительно проверив синтаксис программы с помощью раздела «Помощь»:

```
PROGRAM ASCII;
VAR
   CH:CHAR; {Символ}
   DEC:INTEGER; {Десятичный код символа}
   I,J:INTEGER;
BEGIN
   DEC:=0;
   FOR I:=0 TO 16 DO {16 строк}
     BEGIN
      DEC:=I; {Чтобы получить таблицу кодировки}
              {для символов с кодами 128—255,}
              {этот оператор нужно}
              {заменить на dec:=i+128;}
      FOR J:=1 TO 8 DO {Восемь колонок}
          BEGIN
            IF(DEC<7) OR (DEC>=14)
              THEN
                 WRITE(DEC:4,'–',' ',
                    CHR(DEC):1,CHR(179))
              ELSE {Символы CR,LF,TAB не отображают}
                 WRITE(DEC:4,'–',CHR(179));
            DEC:=DEC+16;
          END;
      WRITELN; {Переход к новой строке экрана}
     END;
END.
```

В результате на экране будет отображена таблица кодов ASCII.

ГЛАВА 3 СРЕДСТВА ИНФОРМАЦИОННЫХ И КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

3.1. История компьютера

Информационные проблемы появились у людей на самой ранней фазе их развития и сопровождают их по сей день. Нашим далеким предкам жизненно важно было знать, например, сколько еды нужно заготовить на случай холодной зимы, как поведет себя тот или иной хищник в разных условиях, какие условия обитания имеются в новых для них местах и т.д. Наверное, они разбирались в понятиях «равно», «не равно», «больше», «меньше», т.е. оперировали информационными признаками предметов, событий.

Из истории известно, что с самых древних времен люди делали попытки как-то облегчить свою умственную деятельность. Например, зарубками, которые делались каменными топорами на стенах пещер, они отмечали количество убитых животных, фактически используя стену пещеры как запоминающее устройство. Огнем зажженных в условленных местах костров, звуками ударов передавали другим информацию о тех или иных событиях, причем на значительные расстояния.

Чем больше развивался человек, тем большее количество информации ему было необходимо помнить, получать, передавать и обрабатывать. Как мы знаем, счет применяется тысячи лет. Первое устройство, механизирующее счет, — счеты — создано задолго до начала нашей эры. Так, разновидность счетов — *абак* — использовалась в Древнем Египте и была знакома китайцам еще в VI в. до н.э. В самых ранних вычислительных устройствах большее внимание уделялось механизации арифметических операций, однако хранение результатов оказалось делом не менее важным. Функции «памяти» выполнял материал — глиняные дощечки, береста, а позднее бумага. Скорость операций была невелика, память использовалась медленно, и, главное,

не было острой необходимости в поиске путей механизации *управления последовательностью операций*.

Прорыв в этой области произошел в ткацком деле. В 1801 г. в ткацком станке Жаккарда использовались *перфокарты* — кусочки картона с пробитыми в них отверстиями для управления командами, обеспечивающими автоматическое переплетение нитей в ткани на станке.

Английский математик Чарльз Бэббидж (1792—1871), исследуя астрономические таблицы, обнаружил в них множество ошибок, вызванных неточностью расчетов. Он понял, что абсолютно точные таблицы можно получить только с помощью расчетов, выполненных машиной, которая получает описание необходимых действий от человека, выполняет их и выдает ему результат. Чарльз Бэббидж выдвинул идею создания программно-управляемой счетной машины, имеющей арифметическое устройство, устройство управления, ввода и печати. Первая спроектированная Бэббиджем машина — *разностная машина* — работала на паровом двигателе (рис. 3.1). Она высчитывала таблицы логарифмов методом постоянной дифференциации и заносила результаты на металлическую пластину. Работающая модель, которую он создал в 1822 г., была шестицифровым калькулятором, способным производить вычисления и печатать цифровые таблицы.

Любопытно, что его невеста, Ада Лавлейс (1815—1852), дочь гениального поэта лорда Байрона, писала программы в кодах этой машины, и поэтому по праву может считаться первым в мире программистом. Имен-

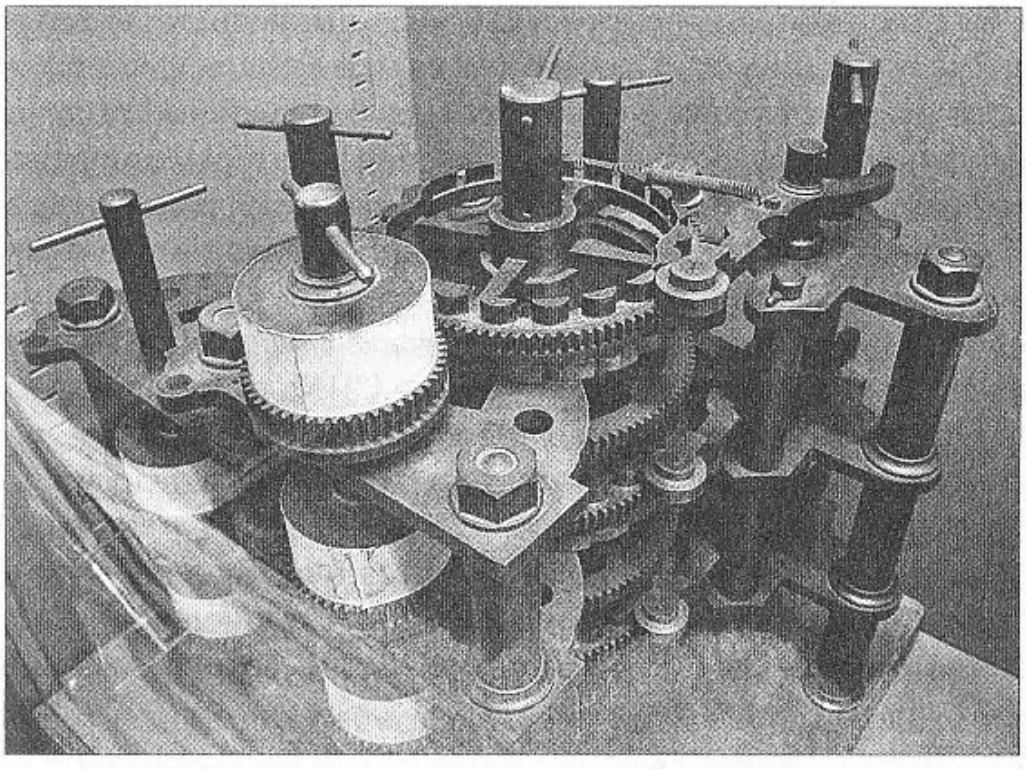


Рис. 3.1. Машина Бэббиджа

но в ее честь разработанный в США в 1960-х гг. язык программирования был назван Ada. В материалах Бэббиджа и комментариях Лавлейс намечены также понятия, как подпрограмма и библиотека подпрограмм, модификация команд и индексный регистр, которые стали употребляться только в 1950-х гг. Сам термин «библиотека» был введен Бэббиджем, а термины «рабочая ячейка» и «цикл» предложила Ада Лавлейс.

В 1844 г. Ада писала о машине Бэббиджа в своей работе «Примечания переводчика»[1]: «Многие лица, недостаточно знакомые с математикой, считают, что роль машины сводится к получению результатов в цифровой форме, а природа самой обработки данных должна быть арифметической и аналитической. Это заблуждение. Машина может обрабатывать и объединять цифровые величины точно так, как если бы они были буквами или любыми другими символами общего характера, и фактически она может выдать результаты в алгебраической форме».

Там же она образно описала значение перфокарт. «Карты только указывают сущность операций, которые должны быть совершены, и адреса переменных, на которые эти действия направлены. Можно сказать достаточно точно, что аналитическая машина ткёт алгебраические удары, как ткацкий станок Жаккарда — цветы и листья».

В 1890 г. в США была проведена перепись населения. Одним из тех, кому была поручена обработка огромного объема информации, полученной в результате этой переписи, был доктор Герман Холлерит. Он придумал специальный код (код Холлерита), который использовал для нанесения на перфокарты для обработки результатов переписи, им же был разработан проект счетной машины, использующей перфокарты. Такая машина была создана, и на ней была решена эта задача. Холлерит развивал успех — появилось целое семейство таких машин разного функционального назначения: перфораторы, контрольники, табуляторы. Они получили название *счетно-перфорационных машин* (СПМ). Эти машины широко использовались в экономике и управлении до 1970-х гг. В нашей стране на всех крупных предприятиях и в учреждениях имелись *машинно-счетные станции* (МСС) или машинно-счетные бюро (МСБ), оснащенные СПМ, на которых решались учетные, бухгалтерские, финансовые и другие задачи.

Первая треть XX в. ознаменовалась созданием и внедрением целого ряда вычислительных устройств. Большой вклад в это внес математик Алан Тьюринг, который в 1937 г. опубликовал работу с ясным и глубоким описанием универсальной схемы вычислений. Для этого он использовал некую гипотетическую, или, говоря сегодняшним

[1] По книге *Апокин И.А.* История вычислительной техники / И.А.Апокин, Л.Е.Майстров. — М. : Наука, 1990.

языком, виртуальную машину с удивительно простой структурой (она так и вошла в историю как «машина Тьюринга»). Правда эта машина так и осталась на бумаге, поскольку из-за своей медлительности уже тогда была экономически невыгодна, но она привлекла к себе внимание многих талантливых ученых и инженеров и, бесспорно, ускорила процесс создания первой универсальной вычислительной машины. Кстати, и сегодня гипотетическая машина Тьюринга и сходная с ней машина Поста используются в теоретической информатике для доказательства теорем, анализа алгоритмов и т. д. Существуют даже компьютерные программы, моделирующие эти машины.

В 1944 г. под руководством Говарда Айкена из компании IBM была создана *электронно-вычислительная машина* Harvard Mark I на релейных логических элементах. В 1946 г. американцы Джон Эккерт и Джон Мочли создали ЭВМ на электровакуумных приборах (электронных лампах) — ENIAC. Обе эти ЭВМ были ориентированы в первую очередь на выполнение научно-технических расчетов. Однако появление технологии ЭВМ **первого поколения** связывают с началом серийного производства UNIVAC в 1951 г. (рис. 3.2).

Нужно сказать, что в нашей стране также было много замечательных достижений в этой области. Так, создание первого *арифмометра* связывают с именем замечательного русского математика П. Л. Чебышева. Что же касается ЭВМ, то в конце 1940-х гг. группа ученых во главе с академиком С. А. Лебедевым создала в Киеве *малую электронно-счетную машину* (МЭСМ), в 1952 г. — *большую электронно-счетную машину* (БЭСМ). Во второй половине 1950-х гг. пошла в серию новая советская ламповая ЭВМ «Урал-1». Чуть позже появилась отечественная ЭВМ «Стрела».

И американские, и наши ЭВМ первого поколения имели десятки тысяч *электронных ламп*, занимали огромные (до 300 м^2) площади и обладали быстродействием в несколько сотен операций в секунду, что по тем временам было просто чудом.

В 1948 г. американские физики Джон Бардин, Уолтер Браттейн и Уильям Шокли изобрели полупроводниковый триод — *транзистор*, что дало толчок к развитию ЭВМ **второго поколения**, где на смену электронным лампам пришли транзисторы, которые имели значительно большую продолжительность службы, более высокую надежность, потребляли меньше энергии, были намного компактнее и гораздо дешевле в производстве. Созданные на базе полупроводников ЭВМ второго поколения, начало массового производства которых относят к 1960 г., имели быстродействие в десятки тысяч операций в секунду, занимали площадь уже не в сотни, а в десятки квадратных метров и гораздо реже, нежели машины первого поколения, выходили из строя.

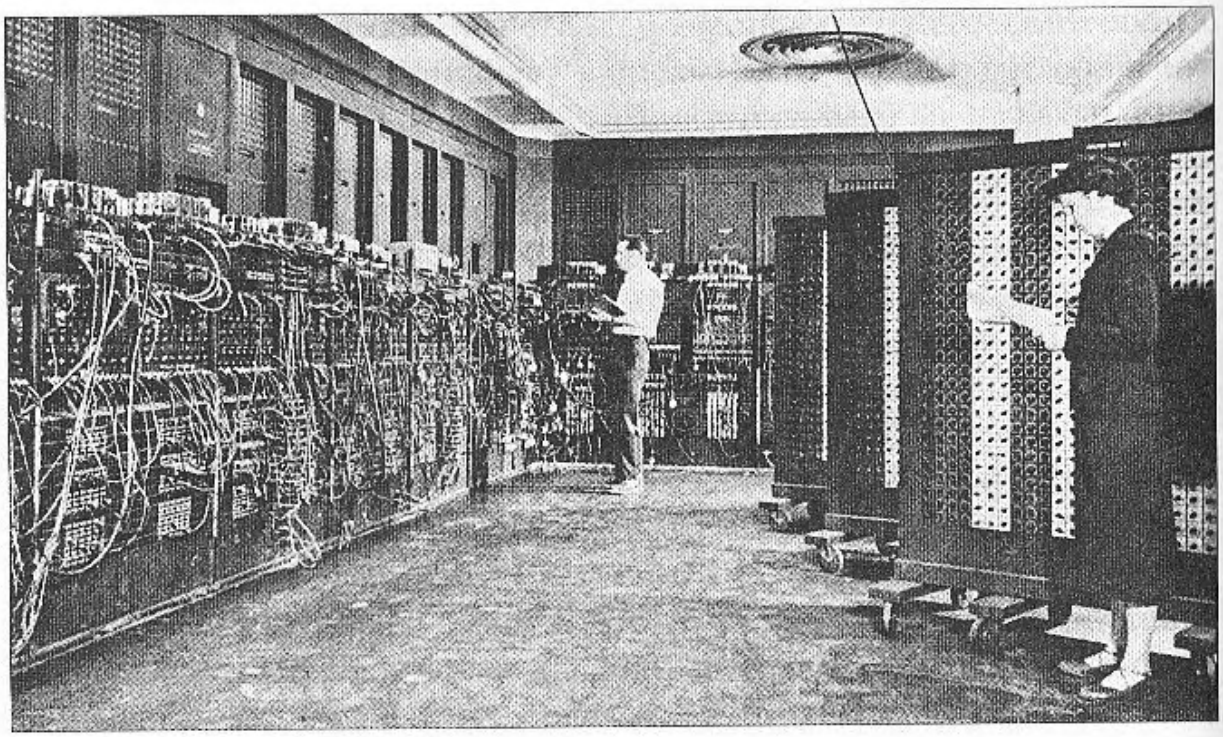


Рис. 3.2. Первый компьютер ENIAC

Наряду с аппаратной составляющей развивалось и программное обеспечение. Дело в том, что для ЭВМ первого поколения, да и в значительной степени второго, программы писались в машинных кодах. Это означает, что у каждого типа ЭВМ была своя система команд, которую программист использовал при написании программы. Сами программы для удобства писались в большинстве случаев в восьмеричной системе счисления, а в машине, естественно, были представлены в двоичной, поскольку каждая восьмеричная цифра имеет соответствующую «триаду» двоичных — от 000 до 111. По своей архитектуре машины подразделялись на одно-, двух- и трехадресные. Например, отечественная ЭВМ «Урал» — одноадресная, «Минск» — двухадресная, а машины типа М-20, как БЭСМ, — трехадресные. Для одно- и двухадресных каждая команда имела модификации: например, для «Минск-22» — первый операнд по первому адресу, второй — по второму, а результат — в сумматор. Или первый операнд — в сумматоре, второй — по адресу, а результат — в адрес памяти и т.д.

В этом смысле более логичными выглядели трехадресные системы команд: код команды (иногда со специальными признаками), первый операнд по первому адресу, второй — по второму, результат записывался по третьему адресу. Например, команда сложения могла выглядеть так: 01 2356 4563 7644. На простом языке это означало: взять число из адреса 2356 оперативной памяти, сложить (код команды — первый в записи) его с числом, находящимся по адресу 4563, а полученный результат записать по адресу, или, как говорили, в ячейку 7644.

Причем команды сложения были тоже разные: с блокировкой нормализации, блокировкой округления, блокировкой того и другого и без блокировок. Кроме того, имелся специальный регистр адреса, содержимое которого программист мог использовать для модификации команд, например при организации циклов. Такие системы команд требовали от программиста не только хорошего знания всех нюансов, но и почти полного представления о работе аппаратуры. При этом возникал интересный момент: программист, пишущий, например, для БЭСМ-4, был беспомощен перед «Минск-22», поскольку система команд, адресация, да и многое другое там было не таким. Это резко снижало эффективность работы. Еще в 1950-х гг. ученые задумывались над тем, как приблизить машинный язык к естественному. Первой важной разработкой в этом направлении было создание символического языка программирования, автором которого была Грейс Хоппер. В этом языке команды заменялись символами, которые мнемонически (Мнемозина — греческая богиня памяти) отражали их суть, адресам можно было давать имена и т.д. Этот язык был ближе человеку, нежели голые коды. Но ведь ЭВМ продолжала работать в кодах, как, впрочем, и сегодня. Потребовалась программа — переводчик с символического языка на язык кодов. Такие программы стали называть *трансляторами*, а впоследствии — *ассемблерами*, а символические языки — *автокодами* или, соответственно, *языками ассемблера*. Естественно, что как сам символический язык (язык ассемблера), так и транслятор (ассемблер) для каждого типа ЭВМ был свой и совершенно неприменимый на другом типе ЭВМ, т.е. эти языки программирования были машинно-ориентированными. Это *языки низкого уровня*, т.е. наиболее приближенные к конкретной системе команд. Все это не давало возможность непосредственно использовать ЭВМ ученым, инженерам, экономистам, и они вынуждены были работать в паре с программистом, чтобы реализовать свои задачи. Такое положение, естественно, долго не могло оставаться неизменным, и уже в 1956 г. появился язык, максимально приближенный к естественному (английскому) языку, который включал в себя многие элементы алгебраических обозначений, — FORTRAN (FORmula TRANslation). В 1960 г. был создан язык Algol-60, который получил большое признание в научном мире, особенно среди математиков, затем появился язык COBOL, предназначенный для решения задач экономики. Языки стали появляться десятками, причем степень их ориентированности на конкретные проблемы (станки с числовым программным управлением, проектирование многослойных печатных плат, управление технологическими процессами и т.д.) возрастала неимоверно. На рубеже 1960–1970-х гг. был создан универсальный язык программирования PL/1

(от англ. Programming Languange — язык программирования). Это был мощный язык, получивший широчайшее распространение в мире и у нас в стране.

Вычислительная техника, естественно, тоже продолжала развиваться. Успехи микроэлектроники, создавшей *микросхемы*, которые включали в себя сотни, а впоследствии и тысячи электронных схем, дали мощный толчок для создания в конце 1960-х гг. ЭВМ **третьего поколения**.

Машины третьего поколения стали выпускать семействами, которые включали в себя малые, средние и большие ЭВМ, причем все они были программно-совместимыми. Самыми известными семействами стали IBM 360 и IBM 370. Они имели унифицированную периферию, были совместимы между собой программно, снизу-вверх, т.е. программы, которые «шли» на более младших моделях семейства, были понятны и выполнимы на более старших моделях. В нашей стране и странах Восточной Европы аналогами IBM 360 и IBM 370 стали семейства «Ряд-1» и, соответственно, «Ряд-2». Например, малую машину ЕС-1010 делали в Венгрии, ЕС-1020 — в Минске, ЕС-1030 — в Казани, ЕС-1040 — в Восточной Германии, ЕС-1050 — в Пензе, ЕС-1060 — в Минске. Аналогичные семейства ЭВМ производились и в других странах — в ФРГ, Японии, Великобритании, Франции.

Важной особенностью ЭВМ третьего поколения явилось появление в них *операционных систем*. Это, конечно, не значит, что подобные попытки не производились ранее: на ЭВМ типа М-20 была так называемая ИС-22 — интерпретирующая система, которая, интерпретируя псевдокод, значительно упрощала вызов стандартных программ на ЭВМ «Минск-32», была программа «Диспетчер», которая, в частности, обеспечивала диалог с пользователем через двухцветную печатающую машинку типа Consul, и т.п. На семействах IBM использовали сначала DOS, а потом OS (Operation System), а также некоторые их разновидности. Именно начиная с ЭВМ третьего поколения благодаря наличию у них ОС был реализован принцип, как тогда говорили, *мультипрограммности* (многозадачности), например в OS могло одновременно существовать до 52 задач, как пользовательских, так и служебных (внутренних). Именно с появлением этих ЭВМ программист был освобожден от исполнения функции управления ресурсами компьютерной системы, это стало главной заботой ОС.

По мере развития микроэлектроники стали появляться ЭВМ **четвертого поколения**, которые строились на основе *больших интегральных схем* (БИС). Процесс микроминиатюризации элементной базы ускорялся и позволил в конце 1980-х гг. приступить к серийному выпуску *персональных компьютеров*.



Рис. 3.3. Ученые — разработчики ЭВМ БЭСМ-6:

слева направо: первый ряд — В. Л. Иванов, С. А. Лебедев, В. И. Семенихин, второй ряд — В. И. Лаут, А. Н. Томилин, Л. Н. Королев, В. М. Смирнов, А. А. Соколов, Л. А. Зак, В. А. Мельников (Москва, СССР, 1969 г., Изофонд ПМ). См. также www.schoolcollection.edu.ru ресурс «Памятники науки и техники в собрании Политехнического музея»

Появились многопроцессорные компьютерные системы, режимы разделения времени, средства телеобработки, телекоммуникации, локальные, а впоследствии и глобальные компьютерные сети, мультимедийные возможности и все то, что так хорошо известно нам сегодня.

Может возникнуть вопрос — зачем нужно знать сегодня о возможностях ЭВМ прошлого? С помощью этих ЭВМ были совершены выдающиеся научные открытия, многие из которых были удостоены Нобелевской премии, осуществлен прорыв в космос, раскрыты секреты атомного ядра, созданы лазерные системы и многое другое. И если вы хотите действительно хорошо разбираться в информатике и в том инструментарии, который она использует, в частности в компьютерной технике, вы должны знать ее историю, поскольку без прошлого нет будущего.

Вы должны знать тех ученых, инженеров, программистов, без гения и труда которых не было бы всего того, чем мы располагаем сейчас в области информатики и компьютерной техники. Все они — дети разных эпох и разных народов — являются гордостью человечества (рис. 3.3).

Проектное задание

Подготовьте для сайта образовательного учреждения материалы «История компьютера». Используйте табл. 3.1, в которой представлены основные поколения развития компьютерной техники.

Таблица 3.1

Поколение, пример	Особенности	Быстродействие (операций в секунду), программное обеспечение
Первое поколение, после 1946 г. ENIAC (США); МЭСМ (СССР)	Применение вакуумно-ламповой технологии, использование систем памяти на ртутных линиях задержки, магнитных барабанах, электронно-лучевых трубках (трубках Вильямса). Для ввода (вывода) данных использовались перфоленты и перфокарты, магнитные ленты и печатающие устройства. Была реализована концепция хранимой программы	10—20 тыс. Машинные языки
Второе поколение, после 1955 г. IBM 701 (США); БЭСМ-6, БЭСМ-4, «Минск-22», «Минск-32» (СССР)	Замена электронных ламп как основных компонентов компьютера на транзисторы. Компьютеры стали более надежными, быстродействие их повысилось, потребление энергии уменьшилось. С появлением памяти на магнитных сердечниках цикл ее работы уменьшился до десятков микросекунд. Главный принцип структуры — централизация. Появились высокопроизводительные устройства для работы с магнитными лентами, устройства памяти на магнитных дисках	100—500 тыс. Алгоритмические языки, диспетчерские системы, пакетный режим
Третье поколение, после 1964 г. IBM 360 (США); ЕС-1030, ЕС-1060 (СССР)	Компьютеры проектировались на основе интегральных схем малой степени интеграции МИС — 10—100 компонентов на кристалл) и средней степени интеграции (СИС — 10—1 000 компонентов на кристалл). Появилась идея, которая и была реализована, — проектирование семейства компьютеров с одной	Порядка 1 млн. Операционные системы (управление памятью, устройствами ввода (вывода) и другими ресурсами), режим разделения времени

Продолжение табл. 3.1

Поколение, пример	Особенности	Быстродействие (операций в секунду), программное обеспечение
	и той же архитектурой, в основу которой положено главным образом ПО. В конце 1960-х гг. появились мини-компьютеры. В 1971 г. появился первый микропроцессор	
Четвертое поколение, после 1975 г. Суперкомпьютеры (многопроцессорная архитектура и использование принципа параллелизма), персональные компьютеры, переносные компьютеры, беспроводные технологии, коммуникаторы в мобильной сети, интерактивный экран, карманный компьютер	Использование при создании компьютеров больших интегральных схем (БИС — 1 000 — 100 000 компонентов на кристалл) и сверхбольших интегральных схем (СБИС — 100 000 — 10 000 000 компонентов на кристалл). Началом данного поколения считают 1975 г. — фирма *Amdahl Corp.* выпустила шесть компьютеров AMDAHL 470 V/6, в которых были применены БИС в качестве элементной базы. Стали использоваться быстродействующие системы памяти на интегральных схемах — МОП-ЗУПВ емкостью в несколько мегабайт. В случае выключения машины данные, содержащиеся в МОП-ЗУПВ, сохраняются путем автоматического переноса на диск. При включении машины запуск системы осуществляется при помощи хранимой в постоянное запоминающее устройство (ПЗУ) программы самозагрузки, обеспечивающей выгрузку ОП и резидентного ОП в МОП-ЗУПВ. В середине 1970-х гг. появились первые ПК	Десятки и сотни миллионов. Базы и банки данных и система управления ими. Пакеты прикладных программ

Окончание табл. 3.1

Поколение, пример	Особенности	Быстродействие (операций в секунду), программное обеспечение
Пятое поколение, после 1982 г. Суперкомпьютеры с мощными функциями искусственного интеллекта	Главный упор при создании компьютеров сделан на их «интеллектуальность», внимание акцентируется не столько на элементной базе, сколько на переходе от архитектуры, ориентированной на обработку данных, к архитектуре, ориентированной на обработку знаний. Обработка знаний — использование и обработка компьютером знаний, которыми владеет человек для решения проблем и принятия решений	Терафлопсы. Операционные системы с графическим интерфейсом, браузеры, web-страницы (сайты), приложения, информационные системы, базы знаний

3.2. Состав персонального компьютера

Современный компьютер — это программно-аппаратный комплекс, который состоит из *аппаратной* (hardware) и *программной* (software) частей. *Архитектура компьютера* обычно определяется совокупностью ее свойств, существенных для пользователя. Основное внимание при этом уделяется структуре и функциональным возможностям компьютера, которые можно разделить на основные и дополнительные.

Основные функции определяют его назначение: обработка и хранение информации, обмен информацией с внешними объектами. *Дополнительные функции* повышают эффективность выполнения основных функций: обеспечивают эффективные режимы работы, диалог с пользователем, высокую надежность и др.

Структура компьютера — это некоторая модель, устанавливающая состав, порядок и принципы взаимодействия входящих в нее компонентов.

Достоинствами ПК являются:

- относительно малая стоимость, находящаяся в пределах доступности для индивидуального покупателя;
- автономность эксплуатации без специальных требований к условиям окружающей среды;



Рис. 3.4. Типовой состав ПК:
ИБП — источник бесперебойного питания

- гибкость архитектуры, обеспечивающая ее адаптивность к разнообразным применениям в сфере управления, науки, образования, в быту;
- «дружественность» ОС и прочего ПО, обусловливающая возможность работы с ней пользователя практически без серьезной специальной профессиональной подготовки;
- высокая надежность работы (более 5 000 ч наработки на отказ).

Состав и назначение основных блоков ПК представлены на рис. 3.4. Организация ПК рассматривается применительно к самым распространенным в настоящее время меркам.

Рассмотрим состав системного блока (рис. 3.5).

Микропроцессор — это центральный узел в системном блоке ПК, предназначенный для управления работой всех блоков компьютера и для выполнения арифметических и логических операций над информацией. В состав микропроцессора входят:

- *устройство управления* — формирует и подает во все блоки компьютера в нужные моменты времени определенные сигналы управления (управляющие импульсы), обусловленные спецификой выполняемой операции и результатами предыдущих операций; формирует адреса ячеек памяти, используемых выполняемой операцией, и передает эти адреса в соответствующие блоки. Опорную последовательность импульсов устройство управления получает от генератора тактовых импульсов;
- *арифметико-логическое устройство* — предназначено для выполнения всех арифметических и логических операций над числовой и символьной информацией;

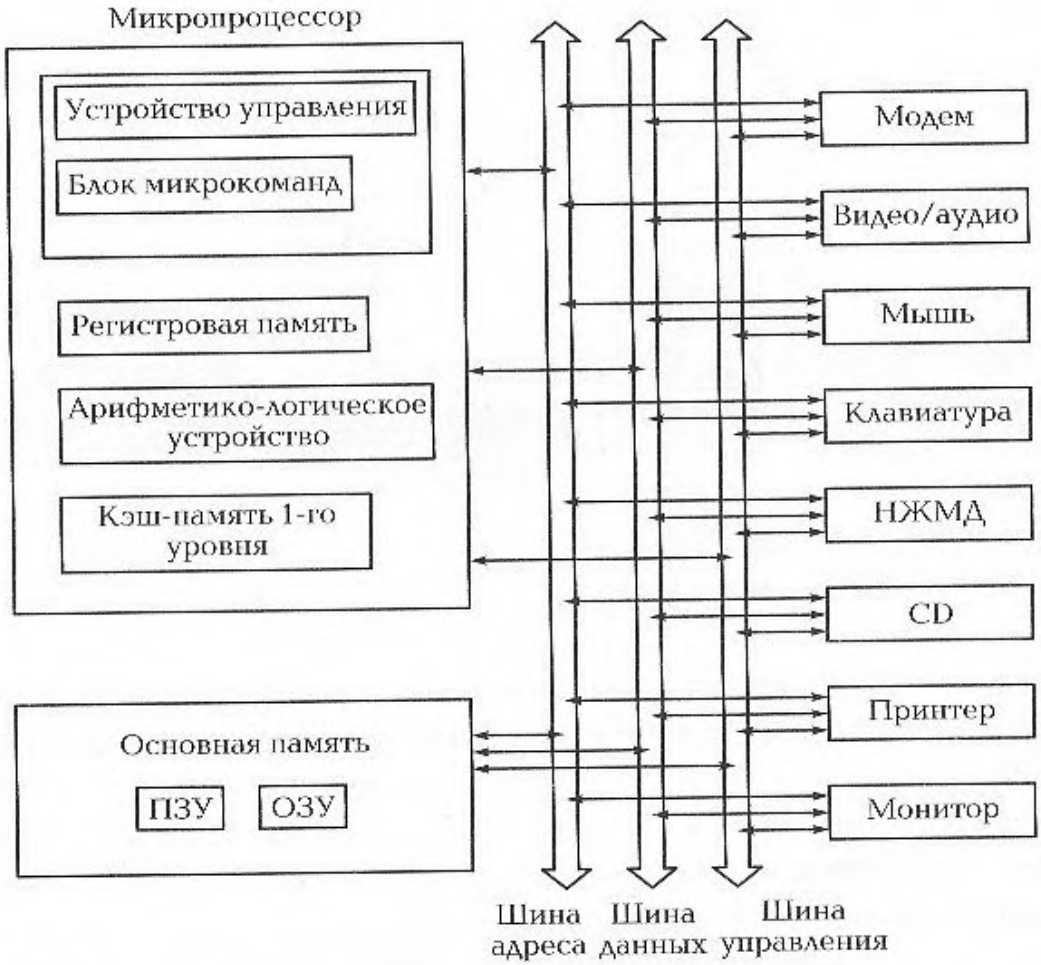


Рис. 3.5. Структурная схема системного блока:
CD — накопитель на компакт-диске

- *микропроцессорная (регистровая) память* — память кратковременного характера для записи и выдачи информации, непосредственно используемой в вычислениях в ближайшие такты работы машины. Регистры — быстродействующие ячейки памяти различной длины (в отличие от ячеек основной памяти (ОП), имеющих стандартную длину 1 байт и более низкое быстродействие);
- *интерфейсная система микропроцессора* — реализует сопряжение и связь с другими устройствами ПК; включает в себя внутренний интерфейс микропроцессора, буферные запоминающие регистры и схемы управления портами ввода (вывода) и системной шиной. Интерфейс — совокупность средств сопряжения и связи устройств компьютера, обеспечивающая их эффективное взаимодействие;
- *порт ввода (вывода)* — аппаратура сопряжения, позволяющая подключить к микропроцессору другое устройство ПК;
- *генератор тактовых импульсов* — генерирует последовательность электрических импульсов; частота генерируемых импульсов определяет тактовую частоту компьютера. Промежуток времени между соседними импульсами определяет время одного такта. Частота

генератора тактовых импульсов является одной из основных характеристик ПК и во многом определяет скорость его работы, ибо каждая операция в машине выполняется за определенное число тактов;

- *системная шина* — основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой.

В свою очередь **системная шина** включает в себя:

а) кодовую шину данных, содержащую провода и схемы сопряжения для параллельной передачи всех разрядов числового кода (машинного слова) операнда;

б) кодовую шину адреса, включающую в себя провода и схемы сопряжения для параллельной передачи всех разрядов кода адреса ячейки основной памяти или порта ввода (вывода) внешнего устройства;

в) кодовую шину управления, содержащую провода и схемы сопряжения для передачи инструкций (управляющих сигналов, импульсов) во все блоки;

г) шину питания, имеющую провода и схемы сопряжения для подключения блоков ПК к системе энергопитания.

Системная шина обеспечивает три направления передачи информации:

1) между микропроцессором и основной памятью;
2) между микропроцессором и портами ввода (вывода) внешних устройств;
3) между основной памятью и портами ввода (вывода) внешних устройств (в режиме прямого доступа к памяти).

Все блоки, а точнее их порты ввода (вывода), через соответствующие унифицированные разъемы — *слоты*, подключаются к шине единообразно: непосредственно или через контроллеры — *адаптеры*. Управление системной шиной осуществляется микропроцессором либо непосредственно, либо, что чаще, через дополнительную микросхему — *контроллер шины*, формирующую основные сигналы управления. Обмен информацией между внешними устройствами и системной шиной выполняется с использованием ASCII- или ANSI-кодов.

Основная память предназначена для хранения и оперативного обмена информацией с прочими блоками. Она содержит два вида запоминающих устройств: постоянное и оперативное. *Постоянное запоминающее устройство* служит для хранения неизменяемой (постоянной) программной и справочной информации, позволяет оперативно только считывать хранящуюся в нем информацию (изменить инфор-

мацию в ПЗУ нельзя). *Оперативное запоминающее устройство* (ОЗУ) предназначено для оперативной записи, хранения и считывания информации (программ и данных), непосредственно участвующей в информационно-вычислительном процессе, выполняемом ПК в текущий период времени. Главными достоинствами оперативной памяти являются ее высокое быстродействие и возможность обращения к каждой ячейке памяти отдельно (прямой адресный доступ к ячейке). В качестве особенности ОЗУ следует отметить невозможность сохранения информации в ней после выключения питания компьютера (энергозависимость). Все перечисленное конструктивно расположено в **системном блоке**.

Источник питания — это блок, содержащий системы автономного и сетевого энергопитания ПК.

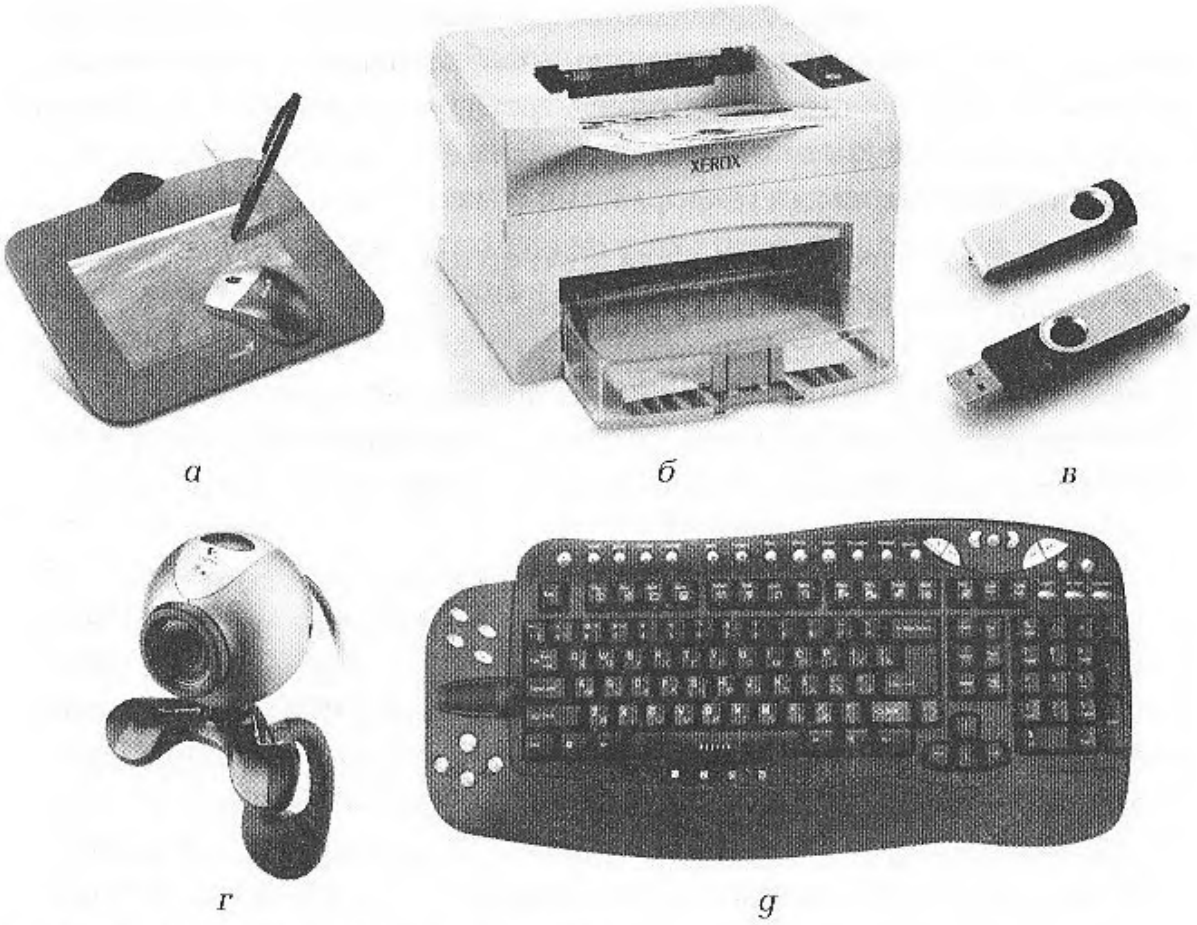
Таймер — это внутримашинные электронные часы, обеспечивающие при необходимости автоматический съем текущего момента времени (год, месяц, часы, минуты, секунды и доли секунд). Таймер подключается к автономному источнику питания — аккумулятору и при отключении машины от сети продолжает работать.

Внешние устройства (ВУ), или *периферия*, — это важнейшая составная часть любого компьютера (рис. 3.6). Достаточно сказать, что по стоимости ВУ иногда могут составлять 50—80 % всего ПК. От состава и характеристик ВУ во многом зависят возможность и эффективность применения ПК в профессии.

Внешние устройства обеспечивают взаимодействие машины с окружающей средой, пользователями, объектами управления и другими компьютерами. Они весьма разнообразны и могут быть классифицированы по ряду признаков. Так, по назначению можно выделить следующие виды ВУ:

- внешние запоминающие устройства, или внешняя память ПК;
- диалоговые средства пользователя;
- устройства ввода информации;
- устройства вывода информации;
- средства связи и телекоммуникации.

Рассмотрим схему на обложке учебника. *Внешняя память* относится к внешним устройствам ПК и используется для долговременного хранения любой информации, которая может когда-либо потребоваться для решения задач. В частности, во внешней памяти хранится все ПО компьютера. Внешняя память содержит разнообразные виды запоминающих устройств, но наиболее распространенными, имеющимися практически на любом компьютере, являются накопители на «жестких» магнитных дисках (НЖМД). Назначение этих накопителей —



а *б* *в*

г *д*

Рис. 3.6. Примеры цифровых устройств:

а — граф-планшет с пером и мышью; *б* — лазерный принтер; *в* — флеш-память; *г* — web-камера; *д* — клавиатура

хранение больших объемов информации, запись и выдача хранимой информации по запросу в ОЗУ. В качестве устройств внешней памяти используются также запоминающие устройства на кассетной магнитной ленте (стримеры), накопители на компакт-дисках (CD-ROM — это Compact Disk Read Only Memory — компакт-диск только для чтения информации на нем) либо пишущие CD-RW (Read-Write), а также флеш-память.

Диалоговые средства пользователя включают в свой состав видеомониторы (дисплеи), манипуляторы типа «мышь» и устройства речевого ввода (вывода) информации.

Видеомонитор (*дисплей*) — устройство для отображения вводимой и выводимой из ПК информации.

Устройства речевого ввода (*вывода*) относятся к быстроразвивающимся средствам мультимедиа.

Устройства речевого ввода — это различные *микрофонные акустические системы*, «звуковые мыши», например со сложным ПО, позволяющим распознавать произносимые человеком буквы и слова, идентифицировать их и закодировать.

Устройства звукового (речевого) вывода — это различные *синтезаторы звука (речи)*, выполняющие преобразования цифровых кодов в буквы и слова, воспроизводимые через громкоговорители (динамики) или звуковые колонки, подсоединенные к компьютеру.

К **устройствам ввода информации** относятся:

- *клавиатура* — устройство для ручного ввода числовой, текстовой и управляющей информации в ПК;
- *графические планшеты* (дигитайзеры) — для ручного ввода графической информации, изображений путем перемещения по планшету специального указателя (пера); при перемещении пера автоматически выполняются считывание координат его местоположения и ввод этих координат в ПК;
- *сканеры* (читающие автоматы) — для автоматического считывания с бумажных носителей и ввода в ПК машинописных текстов, графиков, рисунков, чертежей. В устройстве кодирования сканера в текстовом режиме считанные символы после сравнения с эталонными контурами специальными программами преобразуются в коды, а в графическом режиме считанные графики и чертежи преобразуются в последовательности двухмерных координат;
- *манипуляторы* (устройства указания): джойстик-рычаг, мышь, трекбол в оправе, световое перо и др. — для ввода графической информации на экран монитора путем управления движением курсора по экрану с последующим кодированием координат курсора и вводом их в ПК;
- *сенсорные экраны* — для ввода отдельных элементов изображения, программ или команд с экрана дисплея в ПК.

К **устройствам вывода информации** относятся:

- *принтеры* — печатающие устройства для вывода информации на бумажный носитель, по своим конструкциям разделяются на матричные, струйные и лазерные;
- *графопостроители* (плоттеры) — для вывода графической информации (графиков, чертежей, рисунков) из ПК на бумажный носитель; плоттеры бывают векторные с вычерчиванием изображения с помощью пера и растровые: термографические, электростатические, струйные и лазерные. По конструкции плоттеры подразделяются на планшетные и барабанные. Основные характеристики всех плоттеров примерно одинаковые: скорость вычерчивания 100—1 000 мм/с, у лучших моделей возможны цветное изображение и передача полутонов; наибольшая разрешающая способность и четкость изображения у лазерных плоттеров, но они и самые дорогие.

Средства связи и телекоммуникации предназначены для связи с приборами и другими средствами автоматизации (согласователи интерфейсов, адаптеры, цифроаналоговые и аналого-цифровые преобразователи и т. п.) и для подключения ПК к каналам связи, к другим компьютерам и компьютерным сетям (сетевые интерфейсные платы, сетевые адаптеры, «стыки», мультиплексоры передачи данных, модемы).

В глобальных сетях функции сетевого адаптера выполняет модулятор-демодулятор — *модем*. Многие из названных выше устройств относятся к условно выделенной группе — средствам мультимедиа.

Средства мультимедиа (multimedia) — это комплекс аппаратных и программных средств, позволяющих человеку общаться с компьютером, используя самые разные, естественные для себя среды: звук, видео, графику, тексты, анимацию и др.

К средствам мультимедиа относятся устройства речевого ввода и вывода информации; широко распространенные уже сейчас сканеры (поскольку они позволяют автоматически вводить в компьютер печатные тексты и рисунки); высококачественные видео- (video-) и звуковые (sound-) платы, платы видеозахвата (videograbber), снимающие изображение с видеомагнитофона или видеокамеры и вводящие его в ПК; высококачественные акустические и видеовоспроизводящие системы с усилителями, звуковыми колонками, большими видеоэкранами. Но, пожалуй, еще с большим основанием к средствам мультимедиа относят оптические диски, часто используемые для записи звуковой и видеоинформации — компакт-диски.

Учитывая большую емкость CD (от 650 Мбайт до 4 Гбайт и более), высокую надежность и долговечность, стоимость хранения информации на CD для пользователя оказывается несравнимо меньшей, нежели на магнитных дисках. Это уже привело к тому, что большинство программных средств самого разного назначения поставляется на CD. На компакт-дисках организуются обширные БД, целые библиотеки; на CD представлены словари, справочники, энциклопедии; обучающие и развивающие программы. В чисто бытовом аспекте CD можно использовать для хранения аудио- и видеозаписей, т. е. вместо плеерных аудио- и видеокассет. Следует упомянуть, конечно, и о большом количестве программ компьютерных игр, хранимых на CD.

Мультимедийные проекторы обеспечивают проекцию на большой экран «картинки» с монитора и могут подключаться либо вместо монитора, либо параллельно с ним.

К системной шине и микропроцессору ПК наряду с типовыми внешними устройствами могут быть подключены и некоторые *дополнительные платы* с интегральными микросхемами, расширяющие и улучшающие функциональные возможности микропроцессора: кон-

троллер прямого доступа к памяти, сопроцессор ввода (вывода), контроллер прерываний и др.

Контроллер прямого доступа к памяти освобождает микропроцессор от прямого управления накопителями на магнитных дисках, что существенно повышает эффективное быстродействие ПК. Без этого контроллера обмен данными между внешним и оперативным запоминающими устройствами осуществляется через регистр микропроцессора, а при его наличии данные непосредственно передаются между внешним и оперативным запоминающими устройствами, минуя микропроцессор. *Сопроцессор ввода (вывода)* за счет параллельной работы с микропроцессором значительно ускоряет выполнение процедур ввода (вывода) при обслуживании нескольких внешних устройств (дисплей, принтер, накопители на «жестких» и «гибких» магнитных дисках и др.); освобождает микропроцессор от обработки процедур ввода (вывода), в том числе реализует и режим прямого доступа к памяти. Важнейшую роль играет в ПК *контроллер прерываний*. Прерывание — временный останов выполнения одной программы в целях оперативного выполнения другой, более важной (приоритетной) программы.

Прерывания возникают при работе компьютера постоянно. Достаточно сказать, что все процедуры ввода (вывода) информации выполняются по прерываниям, например, прерывания от таймера возникают и обслуживаются контроллером прерываний 18 раз в секунду (естественно, пользователь их не замечает).

Контроллер прерываний обслуживает процедуры прерывания, принимает запрос на прерывание от внешних устройств, определяет уровень приоритета этого запроса и выдает сигнал прерывания в микропроцессор. Получив этот сигнал, микропроцессор приостанавливает выполнение текущей программы и переходит к выполнению специальной программы обслуживания того прерывания, которое запросило внешнее устройство. После завершения программы обслуживания восстанавливается выполнение прерванной программы.

Как видно из схемы, приведенной на рис. 3.5, современный компьютер строится по *магистрально-модульному принципу*, т.е. к имеющимся магистралям (шинам) можно присоединять через соответствующие устройства (адаптеры, контроллеры) самые разнообразные разноскоростные устройства (модули). С учетом этого говорят об *открытой архитектуре* современного ПК. Открытая архитектура позволяет не только подключать новые устройства, но и модифицировать имеющиеся, наращивать объем оперативной памяти, менять микропроцессор и оперативную память на более производительные и т.д.

Очень важен *принцип программного управления компьютером.* Он заключается в том, что компьютер работает под управлением программ, представляющих собой последовательность команд (инструкций, операций), каждая из которых «понятна» компьютеру и трактуется им однозначно.

В современных компьютерах и программа, и данные, которые обрабатываются ею, находятся в одной оперативной памяти. Этот принцип восходит к самым первым ЭВМ и называется *неймановским принципом* по имени американского ученого Джона фон Неймана, сформулировавшего его.

Мы уже вступили в эпоху цифрового мира — цифровые телевизоры, мобильные телефоны, музыкальные инструменты, видеокамеры — все это совместимо с современными компьютерами и создает некое единое цифровое пространство.

Применение цифровых технологий и соответствующих цифровых устройств в сочетании со специальным ПО позволяет создавать комплексные проекты, включающие в себя программы, рисунки, звук, кино- и видеоизображения, всевозможные звуковые и видеоэффекты, не только для компьютеров, но и в производстве телевизионной, кинопродукции.

Вопросы и задания

1. Из каких конструктивных узлов состоит ПК? В каком из них находится процессор, оперативная память?
2. Каковы функции центрального процессора? Объясните термин «тактовая частота» компьютера.
3. Что такое системная шина (магистраль) компьютера?
4. Что такое порт ввода (вывода)?
5. В чем заключается магистрально-модульный принцип построения компьютера?
6. В чем состоит принцип программного управления?
7. Какие устройства называют мультимедийными и почему?
8. Назовите основные устройства ввода и вывода информации.
9. Какие носители предназначены для длительного хранения информации?
10. Что такое цифровые технологии?

Проектное задание

Для сайта образовательного учреждения подготовьте материал «Цифровые технологии» и «Компьютер и профессия». Опишите состав и оцените стоимость ПК автоматизированного рабочего места для выбранной вами профессии. Для подбора цифровых устройств и компьютера воспользуйтесь сайтами интернет-магазинов компьютерной техники.

3.3. Логические функции и схемы — основа элементной базы компьютера

В жизни мы часто, сами того не замечая, используем логические зависимости в самых разных ситуациях. Например, мы говорим: «чтобы приготовить это блюдо, мне необходимо обязательно иметь мясо, соус, соль и зелень» (перефразируя, можно сказать, что нужны «И мясо, И соус, И соль, И зелень»). При этом подразумевается, что отсутствие даже чего-либо одного из перечисленного не даст возможности приготовить именно это блюдо. Другой пример: направляясь в магазин, вы знаете, что можете доехать до него ИЛИ на трамвае, ИЛИ на автобусе, ИЛИ на троллейбусе. В принципе, все равно, на чем ехать, не устраивает только одна ситуация, при которой не будет ни трамвая, ни автобуса, ни троллейбуса. Еще пример: собираясь с друзьями в поход, вы предполагаете «надену любую обувь, только НЕ эти кроссовки, поскольку они неудобны».

А еще мы часто увязываем наступление или ненаступление каких-то событий в более сложные логические формы. Например, «я пойду гулять, если успею выполнить домашние задания и если на улице не будет дождя», «чтобы добраться до стадиона, нужно сначала ехать на метро, а потом на автобусе или троллейбусе», «если в выходной будет морозец, то я пойду с ребятами кататься на лыжах, иначе (т.е. если будет слякоть) останусь дома и займусь компьютером».

Здесь мы привели простые, бытовые, но тем не менее логические зависимости. Несложно сообразить, что при обработке информации такие зависимости возникают сплошь и рядом.

Привычная алгебра оперирует числами, а алгебра логики — событиями. *Алгебру логики* иначе называют *булевой* (в честь ее создателя английского математика Джорджа Буля) или *алгеброй высказываний*.

Чтобы понимать дальнейшее, вы должны узнать одно очень полезное *правило* — любая формальная математическая система в общем случае состоит из следующих основных множеств:

- операндов (данных для обработки, или, иными словами, информации);
- операций (операторов или действий над информацией);
- постулатов (законов, правил, теорем, аксиом, формул, по которым осуществляется процесс обработки информации).

Этому правилу подчиняется и алгебра логики. Нас будет интересовать так называемый двузначный вариант алгебры логики, которая в качестве множества элементов содержит два элемента «0» и «1». Будем называть их *константами алгебры логики*, и, чтобы не путать

с двоичными цифрами, говорить о логическом «0» и логической «1». Именно по причине использования этих двух констант алгебру логики еще называют *переключательной алгеброй*, поскольку типовой переключатель имеет два фиксированных положения.

Условимся, что если некоторое событие A произошло, то это записывается как $A = 1$, а если оно не произошло, то $A = 0$. В любом случае

$$A = 0, \text{ если } A \neq 1;$$
$$A = 1, \text{ если } A \neq 0.$$

Для обозначения событий, или как мы будем их называть в дальнейшем, булевых переменных, используем буквы латинского алфавита. Таким образом, мы определили множество элементов алгебры логики.

Рассмотрим наиболее распространенные логические операции.

Логическое умножение (конъюнкция) или операция И. Давайте сначала попробуем разобраться, что называется «на пальцах». Представьте себе елочную гирлянду из 10 лампочек, которые включены последовательно. Если хотя бы одна лампочка перегорит, то ток прекратится и гирлянда гореть не будет. Следовательно, событие A (гирлянда горит) наступит тогда и только тогда, когда произойдут события B_1, B_2, B_3, ..., B_{10}, где соответственно B_1 — исправность первой лампочки, B_2 — второй и т.д. Сколько может быть комбинаций различных состояний? Мы с вами уже знаем, как это определить. Каждый элемент (лампочка) может иметь два состояния (исправна-неисправна), а всего таких элементов 10, следовательно, количество комбинаций определим как

$$2^{10} = 1\,024.$$

Иными словами, из 1 024 комбинаций событий «исправна-неисправна лампочка» только при одной комбинации, когда все лампочки исправны, т.е. $B_1 = 1$, $B_2 = 1$, $B_3 = 1$, ..., $B_{10} = 1$, произойдет событие A, т.е. $A = 1$, — гирлянда будет гореть. Во всех остальных 1 023 случаях, если хотя бы одно из событий B_1, B_2, ..., B_{10} не наступит, т.е. будет равным 0, гирлянда гореть не будет, — событие A не наступит, т.е. $A = 0$.

Другой пример. Думаем, что вы согласны с тем, что урок информатики в вашей группе (событие S) состоится, если произойдет событие A (пришел класс) и произойдет событие B (пришел учитель). Очевидно, что S — это событие-функция, наступление которого зависит от наступления событий A и B (события-аргументы). Сколько может быть комбинаций значений событий аргументов и, соответственно, значений события-функции? Конечно, четыре (табл. 3.2).

Таблица 3.2

A	*B*	*S*
0 (класс не пришел)	0 (учитель не пришел)	0 (урок не состоялся)
0 (класс не пришел)	1 (учитель пришел)	0 (урок не состоялся)
1 (класс пришел)	0 (учитель не пришел)	0 (урок не состоялся)
1 (класс пришел)	1 (учитель пришел)	1 (урок состоялся)

Обобщая, можно сказать, что событие *S* произойдет только в том случае, если произойдут и событие *A*, и событие *B*. Во всех остальных случаях оно не произойдет. Вот почему логическое умножение иногда называют операцией И. Эту операцию обозначают символом «&», а иногда и символом «*».

Запишем таблицу значений логической функции $S = A*B$ (табл. 3.3).

Если мы рассмотрим логическое выражение $A = B*C*D*E*F$, то нетрудно сообразить, что общее количество комбинаций значений логических операндов *B*, *C*, *D*, *E*, *F* будет равно 32, но только в одном случае, когда все они будут равны 1, значение логической функции *A* будет 1, во всех остальных 31 случаях $A = 0$.

Итак, можно подметить *закономерность, свойственную конъюнктивным формам* (так мы будем называть логические выражения, состоящие из операций конъюнкции): в конъюнктивных формах значение логической функции равно 1 только в одном случае — когда все логические операнды равны 1, во всех остальных случаях значение логической функции равно 0. Эта закономерность очень пригодится нам при упрощении сложных логических выражений. Действительно, стоит убедиться, что один из операндов конъюнктивной формы равен 0, и дальше можно не считать, а быть уверенным, что и значение логической функции равно 0.

Таблица 3.3

A	*B*	$S = A*B$
0	0	0
0	1	0
1	0	0
1	1	1

Логическое сложение (дизъюнкция) или операция ИЛИ. Рассмотрим пример. В вашей комнате висит светильник, состоящий из трех лампочек, каждая из которых может включаться отдельно. Когда произойдет событие *S* — в комнате горит

Таблица 3.4

A	*B*	*S*
0 (дождь не идет)	0 (снег не идет)	0 (осадков нет)
0 (дождь не идет)	1 (снег идет)	*S*1 (осадки есть)
1 (дождь идет)	0 (снег не идет)	1 (осадки есть)
1 (дождь идет)	1 (снег идет)	1 (осадки есть)

свет? Очевидно, тогда, когда будет гореть или одна лампочка, или любые две, или все три. И только в одном случае из восьми ($2^3 = 8$) событие *S* не наступит — когда ни одна из лампочек гореть не будет.

Другой пример. Стоит поздняя осень. Рассказывая по телевизору о прогнозе погоды, сотрудник Гидрометеоцентра говорит: «Завтра возможны осадки». Когда это событие будет истинным? Очевидно, что тогда, когда будет идти снег или дождь, или, что бывает тоже, снег с дождем. И только в одном случае это событие не произойдет — когда ни снега, ни дождя, ни снега с дождем не будет. Обозначим событие «выпадение осадков» как *S*, дождь — как *A*, снег — как *B*. Как вы понимаете, комбинаций аргументов может быть четыре ($2^2 = 4$).

Обобщая, заметим, что событие *S* произойдет, если произойдет или событие *A*, или событие *B*, или они оба. И только в одном случае событие *S* не произойдет — когда не произойдет ни одно из событий-аргументов (табл. 3.4).

Именно поэтому логическое сложение иногда называют операцией ИЛИ. Эту операцию обозначают символом «∨», а также символом «+».

Запишем значение логической функции $S = A + B$ (табл. 3.5).

Таблица 3.5

A	*B*	$S = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Если же мы рассмотрим более сложную *дизъюнктивную форму*, например: $A = B + C + D + E + F + G$, то сразу определим, что количество комбинаций значений логических операндов *B*, *C*, *D*, *E*, *F*, *G* будет равно 64, так как каждый операнд может иметь два значения, а всего таких операндов шесть, причем в 63 случаях значение функции будет равно 1 и только в одном 0. Попробуем обобщить.

Таблица 3.6

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

В дизъюнктивных формах значение логической функции равно 0 только в одном случае — когда все логические операнды равны 0, во всех остальных случаях значение логической функции равно 1. Эта закономерность очень полезна при упрощении сложных логических выражений. Действительно, стоит убедиться, что один из операндов дизъюнктивной формы равен 1, как вывод очевиден — значение логической функции равно 1.

Логическое отрицание (инверсия) или операция НЕ. Смысл этой функции заключается в том, что если значение логической переменной есть $A = 1$, то значение ее отрицания есть $A = 0$, и наоборот (табл. 3.6).

Логические схемы составляют основу построения многих технических систем обработки информации, в том числе, конечно, и компьютеров. Нас они будут интересовать как «черный ящик», т. е. нам важно знать, что логические схемы получают на входе, и что — на «выходе». Логические схемы выполняются, как правило, из диодов и (или) транзисторов и их работа описывается алгеброй логики.

На рис. 3.7 представлена *логическая схема И* (логическое сложение, конъюнкция). Схема имеет входы A и B и выход S. На вход подаются сигналы, соответствующие значениям аргументов. Например, если $A = 1$, то на входе A есть сигнал, если же $B = 0$, то на входе B сигнала нет. Что такое сигнал? В общем случае это некое физическое воздействие на вход схемы. В реальных схемах это, как правило, воздействие электрическим током (напряжением). Очевидно, что приведенная логическая схема реализует операцию $S = A * B$ и, следовательно, будет работать в соответствии с приведенной ранее таблицей значений, т. е. сигнал на выходе S будет только в том случае, когда будут сигналы на входе A и на входе B, в остальных трех случаях сигнала на выходе не будет, т. е. $S = 0$.

Давайте придумаем ситуацию, в которой эта схема может быть применима. Представьте себе, что мы задумали экономить электроэнергию на внутреннем освещении подъездов, чтобы лампочки загорались только тогда, когда света из окон уже недостаточно и когда в подъезде находится человек. Очевидно, что такая схема будет работать следующим образом: на окнах установлены фотодатчики, в которых

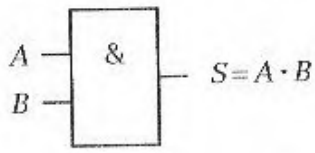


Рис. 3.7. Логическая схема И

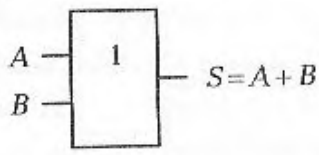


Рис. 3.8. Логическая схема ИЛИ

возникает сигнал при уровне освещенности на улице ниже установленной, этот сигнал подается на вход A, в то же время при нахождении человека в подъезде (на этаже) срабатывает фотоэлемент, сигнал с которого передается на вход B, на выходе S в этом случае возникает сигнал, который и зажигает внутренне освещение подъезда. Нетрудно догадаться, что при достаточном уровне освещенности на улице или при отсутствии людей в подъезде, или в ситуации, когда имеет место и то и другое, освещение в подъезде включаться не будет.

Другой пример. Имеется закрытая стоянка на 50 автомашин. Для того чтобы подъезжающий водитель мог издалека получить информацию о наличии свободных мест, сделаем следующее. В пол каждого бокса установим датчик, который срабатывает под тяжестью стоящей автомашины, этот сигнал подается на схему И, имеющую 50 входов. Очевидно, что сигнал на выходе схемы возникает только в том случае, когда сигналы будут на всех входах, и выходной сигнал зажжет подсветку табло с надписью «Мест нет», во всех остальных случаях это табло гореть не будет.

Рассмотрим *логическую схему ИЛИ* (логическое сложение, дизъюнкцию) (рис. 3.8).

Схема имеет входы A и B и выход S. На вход подаются сигналы, соответствующие значениям аргументов. Например, если $A = 1$, то на входе A есть сигнал, если же $B = 0$, то на входе B сигнала нет. О том, что такое сигнал, мы уже говорили. Поскольку приведенная логическая схема реализует операцию $S = A + B$, она будет работать в соответствии с приведенной ранее таблицей, т. е. сигнал на выходе S будет тогда, когда на одном из входов A и B или же на обоих будет сигнал. Когда же на входах не будет сигнала, его не будет и на выходе S.

Рассмотрим примеры, описываемые дизъюнктивными закономерностями. Предположим, что есть трубопровод, схема которого приведена на рис. 3.9.

Жидкий продукт, например нефть, транспортируется из пункта 1 в пункт 2. На трассе трубопровода имеются разветвления, проходящие через населенные пункты A, B и C, в которых имеются нефтераспре-

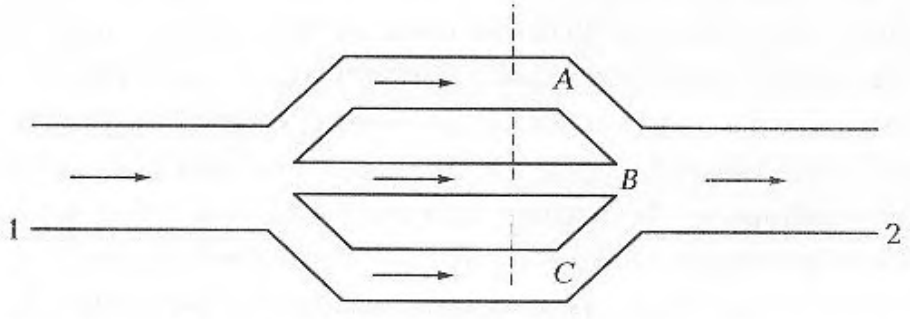


Рис. 3.9. Схема «трубопровода»

Таблица 3.7

A	B	C	$S = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

делительные станции, снабженные вентилями, способными перекрыть движение нефти по трубе. Условимся, что если вентиль в пункте A открыт, то $A = 1$ и т. п. Когда произойдет событие S — поступление нефти в пункт 2? Очевидно, что возможны восемь комбинаций значений аргументов логической функции $S = A + B + C$ (табл. 3.7).

Из табл. 37 видно, что из восьми возможных комбинаций открыт-закрыт вентилей A, B и C только при одной из них, а именно когда все три вентиля будут перекрыты, нефть в пункт 2 поступать не будет, т. е. событие $S = 0$. Допустим, что на центральном пульте управления трубопроводом имеется индикатор, который подключен к выходу схемы ИЛИ, на вход схемы с каждого вентиля подается сигнал, если вентиль открыт. Очевидно, что индикатор (лампочка) будет светиться во всех случаях, кроме одного — когда все три вентиля будут закрыты.

Рассмотрим на примере *логическую схему НЕ*. Допустим, что у нас есть обогреватель воздуха, который мы хотим заставить работать в автоматическом режиме. Это можно сделать так: с датчика температуры сигнал поступает в том случае, если температура в комнате в норме. Поступает он на специальную схему, которую будем называть схемой НЕ или *инвертором*. Эта схема работает так, что если на входе есть сигнал, то на выходе — нет, и наоборот, если сигнала нет на входе, то на выходе он есть. Итак, пока температура в помещении в пределах установленной нормы, сигнал от датчика поступает на схему НЕ, и на

ее выходе, к которому подключен обогреватель, сигнала, включающего его, нет. Как только температура опустится ниже установленной, сигнал от датчика на входе схемы НЕ пропадет, а сигнал на ее выходе появится и включит обогреватель.

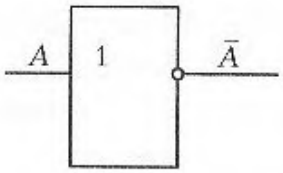


Рис. 3.10. Логическая схема НЕ

Будем обозначать логические схемы, реализующие функцию НЕ так, как показано на рис. 3.10.

«Исключающее ИЛИ». Операция сложения по mod2. Эта логическая операция также заслуживает внимания, поскольку она достаточно широко используется в качестве внутренней команды микропроцессоров. Она обозначается знаком «⊕». Иллюстрирует эту операцию табл. 3.8.

Закономерность очевидна: если операнды одинаковые, то результат равен 0; если операнды разные, то результат равен 1.

Приведем пример использования этой операции в компьютерных микропроцессорах. *Сумматор* — вид регистра, обладающий способностью сложения двоичных кодов. Сумматор — неотъемлемая и основная часть любого процессора. Пусть в нем находится некоторое восьмибитовое число A, в регистре — восьмибитовое число B. Сложим содержимое регистра с содержимым сумматора по mod2 (логическая операция «Исключающее ИЛИ») (рис. 3.11).

Числа A и B были неизвестны, но результат их побитового (поразрядного) сложения оказался равен 00000000. О чем это говорит? Видимо, о том, что числа были одинаковые. И наоборот, если результат такого сложения будет ненулевым, если хотя бы в одном разряде будет 1, это говорит о том, что числа разные и отличаются они именно теми разрядами, в которых после выполнения операции «Исключающее

Таблица 3.8

A	B	$S = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

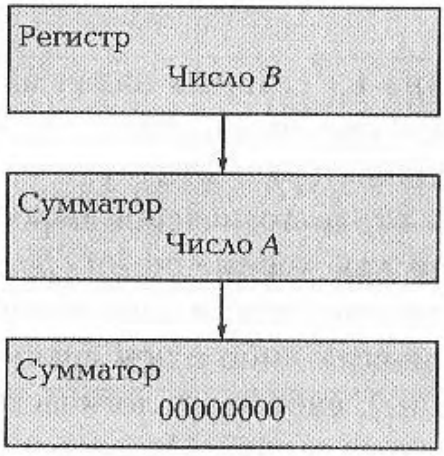


Рис. 3.11. Схема сложения двух чисел по mod2 с использованием сумматора

ИЛИ» образовались единицы. Кстати, именно так компьютерный процессор *сравнивает числа*. Вырабатываемый (или невырабатываемый) сигнал «0» (сумматора) как раз и несет информацию об этом.

? Вопросы и задания

1. В чем состоит принципиальное отличие обычных алгебраических операций от логических (булевских)?
2. Какие логические операции вы знаете? Как они обозначаются? Приведите примеры жизненных ситуаций, которые содержат логические зависимости.
3. С чем в отличие от обычной алгебры оперирует алгебра логики? Почему алгебру логики иначе называют переключательной алгеброй?
4. Как в алгебре логики обозначают наступление события *A*? Как в алгебре логики обозначают ненаступление события *B*? Может ли событие *C* одновременно обозначаться 0 и 1?
5. Приведите примеры бытовых ситуаций, которые можно описать операцией логического умножения (конъюнкцией), логического сложения (дизъюнкцией). Придумайте случай, когда для автоматизации какого-либо процесса целесообразно использовать логическую схему И, логическую схему ИЛИ.

П-Т Проектное задание

В гостиницах, аэропортах есть двери, которые открываются сами при приближении человека. Опишите логическую схему, которую можно использовать для реализации логической функции «человека нет — дверь закрыта, человек есть — дверь открыта».

3.4. Логические выражения и таблицы истинности. Логические схемы и логические диаграммы

Одна из главных задач алгебры логики — описание поведения и структуры логических схем. Рассмотрим логическую схему как «черный ящик» (рис. 3.12).

На входе логические переменные (*X1*, *X2*, *X3*, ..., X_n). На выходе логическая переменная *f*. Для того чтобы описать поведение этого «черного ящика», нужно выход выразить через вход. Это можно сделать двояко: либо с помощью таблицы истинности (комбинационной таблицы), либо через логическое выражение. Строго говоря, возможна и обратная задача. Например, имеется логическое выражение, описывающее какую-либо ситуацию, закономерность, событие, и необходимо построить модель в виде логической схемы, которая представляет эту

ситуацию, закономерность, событие. Кстати, это очень интересная задача, поскольку, варьируя входом, можно анализировать, к чему приведет та или иная комбинация входных значений. Думаем, что вы сами можете привести десятки таких примеров.

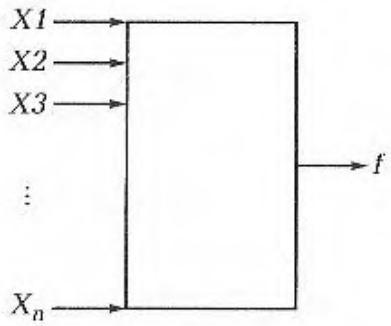


Рис. 3.12. Представление логической схемы в виде «черного ящика»

Логическая схема, которую можно полностью описать таблицей истинности или логическим выражением, называется *комбинационной*. В таких схемах значения входных переменных в текущий момент времени полностью определяют значения выходных переменных. Однако есть и другой класс логических схем — это *схемы с внутренней памятью*, их еще называют *последовательностными*. В этих схемах значения выходных переменных определяются не только значениями входных переменных, но и их значениями в предыдущие моменты времени. Здесь мы эти схемы не рассматриваем, ограничимся лишь их упоминанием и будем знать о том, что такие схемы есть.

Вернемся к нашему «черному ящику». Как вы понимаете, каждая из входных переменных может принимать только одно из двух значений. Все точки внутри «черного ящика» обладают тем же свойством, впрочем, как и выход. *Таблица истинности* (*комбинационная таблица*) — это таблица, которая содержит все комбинации входных переменных (аргументов) вместе с соответствующими им значениями выходных переменных (функций).

При n входных и одной выходной переменных таблица содержит 2^n строк и $n + 1$ столбцов значений (табл. 3.9).

Таблица 3.9

X1	*X2*	...	X_n	Значение	*f*
0	0	...	0	0	*f*(0, 0, ..., 0, 0)
0	0	...	0	1	*f*(0, 0, ..., 0, 1)
0	0	...	1	0	*f*(0, 0, ..., 1, 0)
0	0	...	1	1	*f*(0, 0, ..., 1, 1)
...	...	...	...	...	...
1	1	...	1	1	*f*(1, 1, ..., 1, 1)

Самый простой способ указания всех возможных значений входных величин — двоичный перебор всех значений от 0 до $2^n - 1$. Очевидно, что f будет принимать для каждой комбинации аргументов значения либо логический 0, либо логическая 1 в зависимости от заданной логической функции.

Еще один способ — задать *логическое выражение*. Оно представляет собой формулу, состоящую из логических констант и переменных, связанных операциями НЕ, И, ИЛИ. Как и в обычных алгебраических выражениях, действует правило скобок. Приоритет выполнения логических операций соответствует порядку их записи в скобках. При этом логическое умножение имеет приоритет над логическим сложением. Далее приведен пример логического выражения, описывающего логическую функцию от трех аргументов:

$$f(X1, X2, X3) = (X1 + \overline{X2})(\overline{X1} + X3) + X2X3.$$

Построить таблицу истинности по логическому выражению достаточно просто: вычисляется значение функции для каждой комбинации входных значений. Для примера возьмем логическую функцию, приведенную выше. Очевидно, что поскольку аргументов три, то количество значений функции будет $2^3 = 8$. Это значит, что в нашей таблице будет восемь строк (табл. 3.10).

Таблица 3.10

$X1$	$X2$	$X3$	$\overline{X1}$	$\overline{X2}$	$X1+\overline{X2}$	$\overline{X1}+X3$	$(X1+\overline{X2})(\overline{X1}+X3)$	$X2X3$	f
0	0	0	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	0	0	0
0	1	1	1	0	0	1	0	1	1
1	0	0	0	1	1	0	0	0	0
1	0	1	0	1	1	1	1	0	1
1	1	0	0	0	1	0	0	0	0
1	1	1	0	0	1	1	1	1	1

Для того чтобы построить такую таблицу истинности, нужно вычислить значение функции f для каждой комбинации аргументов. Для примера рассмотрим одну из комбинаций: 0, 1, 1. Тогда

$$f(0, 1, 1) = (0 + \bar{1})(0 + 1) + 1 * 1 =$$
$$= (0 + 0)(1 + 1) + 1 = 0 * 1 + 1 = 0 + 1 = 1.$$

Существуют и другие процедуры построения таблицы истинности по логическому выражению, но мы ограничимся рассмотренной.

Логические схемы — это схемы, из которых собственно и состоит *элементная база компьютера*. Алгебра логики позволяет описать их поведение и структуру. Однако до сих пор мы рассматривали только поведенческий аспект, т. е. и логические выражения, и таблицы истинности использовались для описания выходных величин в зависимости от входных.

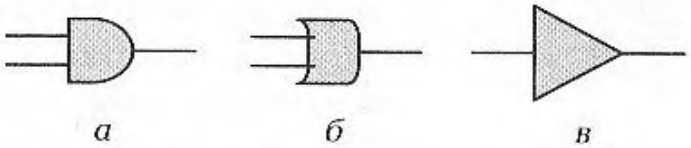
Мы уже рассмотрели три логические схемы, соответствующие трем основным операциям алгебры логики, — И, ИЛИ и НЕ. Из этих схем, как вы уже поняли, можно формировать более сложные схемы, в которых логическая зависимость выхода от входа будет описываться логическим выражением. Такие логические схемы принято называть *вентильными схемами* или *вентилями*. Вентиль НЕ еще называют *инвертором*. Обычный водопроводный кран на языке сантехников тоже называется вентилем, т. е. устройством, пропускающим или не пропускающим жидкость. У нас, конечно, на входах и выходах — электрические сигналы, однако от их характеристик можно абстрагироваться и условно считать, что на входах и выходах имеются непосредственно логические значения — «0» или «1». Покажем вид основных логических схем (рис. 3.13).

Допустим, имеется логическая функция

$$f(W, X, Y, Z) = W(XYZ + \bar{Y}\bar{Z}).$$

Построим логическую схему, которая описывает поведение этой функции (рис. 3.14).

Рисунки, которые состоят из элементарных логических схем, соединенных связями, называют *логическими диаграммами*. В общем случае, когда логическая диаграмма состоит из вентилей и не содержит обратных связей, она называется *комбинационной*. Комбинационная ло-



а *б* *в*

Рис. 3.13. Обозначение логических схем:
а — схема И; *б* — схема ИЛИ; *в* — схема НЕ

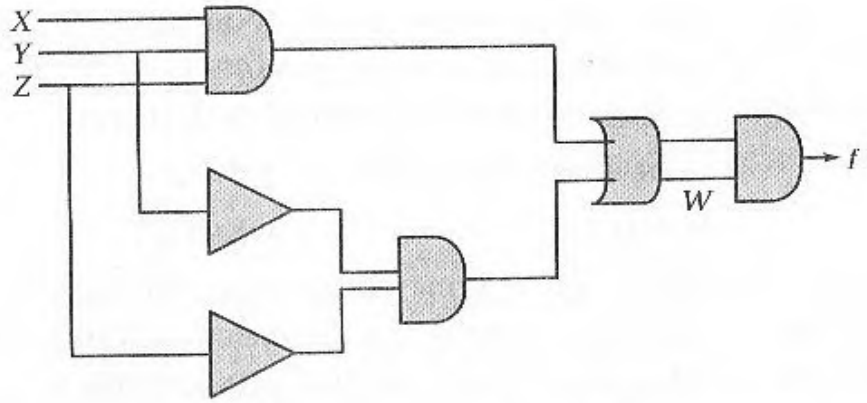


Рис. 3.14. Логическая диаграмма, реализующая конкретную логическую функцию

гическая диаграмма не обладает запоминающими свойствами, поэтому значения на ее выходах полностью определяются значениями на входах в данный момент времени. Вам уже понятно, что существует однозначное соответствие между диаграммами комбинационных схем и булевыми выражениями. Нужно сказать, что каждый аргумент подается на вход только один раз, т.е. если диаграмма реализует логическую функцию от четырех аргументов, значит, у нее должно быть четыре входа.

Для примера рассмотрим последнюю приведенную комбинационную логическую диаграмму. У нее четыре входа: X, $\overline{Y}$, Z и W. Два инвертора используются для получения значений $\overline{Y}$ и $\overline{Z}$, выход левого верхнего вентиля «И» соответствует XYZ, а выход левого нижнего вентиля «И» дает $\overline{Y}\,\overline{Z}$. Выходы обоих этих вентилей служат входами вентиля «ИЛИ», поэтому на его выходе будет $XYZ + \overline{Y}\,\overline{Z}$. Наконец, выход этого вентиля подается на вход другого вентиля «И», на другой вход которого подается W. Окончательно эта логическая диаграмма описывается следующей функцией:

$$f(W, X, Y, Z) = W(XYZ + \overline{Y}\,\overline{Z}).$$

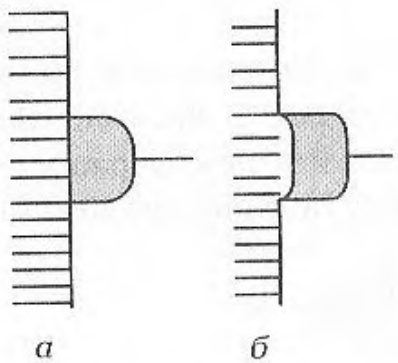
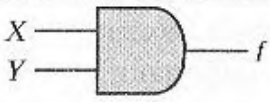
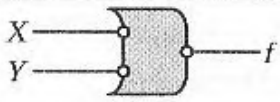
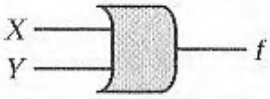
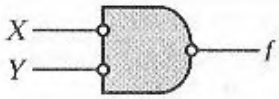
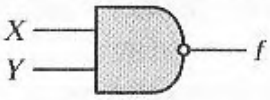
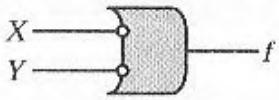
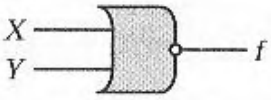
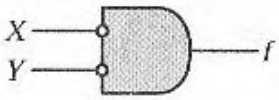


Рис. 3.15. Многовходовые вентили:
а — вентиль И; *б* — вентиль ИЛИ

Очевидно, что можно осуществить и обратный процесс, т.е. по заданному булевому выражению построить соответствующую логическую диаграмму. Все это, конечно, не значит, что вентили «И» и «ИЛИ» могут иметь только по два или три входа, их может быть гораздо больше, выход у них, естественно, один. Только лишь инвертор имеет один вход и один выход — в этом его суть. На рис. 3.15 представлены такие «многовходовые» вентили «И» и «ИЛИ».

Таблица 3.11

Функция	Графическое обозначение вентиля		Булево выражение
	прямой	инверсный	
И			$f = XY = (\overline{\overline{X} + \overline{Y}})$
ИЛИ			$f = (X + Y) = (\overline{\overline{X}\,\overline{Y}})$
И–НЕ			$f = (\overline{XY}) = \overline{X} + \overline{Y}$
ИЛИ–НЕ			$f = (\overline{X + Y}) + \overline{XY}$

В данном подразделе рассмотрены вентили только трех основных типов, однако в логических диаграммах используют и другие типы вентилей, многие из которых представляют собой комбинации из основных.

В табл. 3.11 приведены графические обозначения вентилей наиболее распространенных типов. Отметим, что, во-первых, в этой таблице представлено несколько новых функций и, во-вторых, что каждая функция представлена парой графических изображений. В этой системе кружочек обозначает инверсию.

Как вы сами хорошо понимаете, приведенными моментами данная тема далеко не исчерпывается. Это целый мир — увлекательный, интересный, порою сложный, но удивительный и огромный.

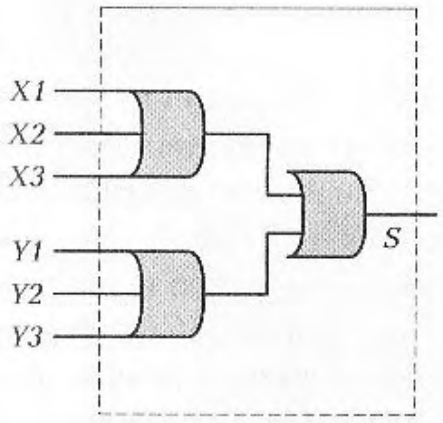


Рис. 3.16. Пример логической схемы из трех элементов ИЛИ

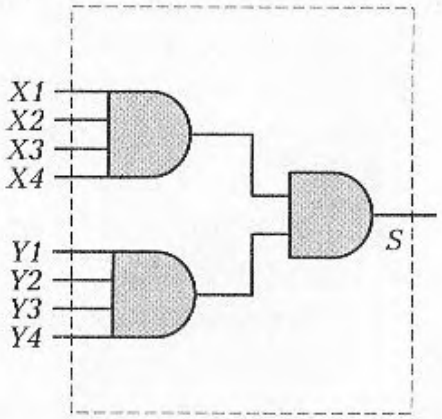


Рис. 3.17. Пример логической схемы из четырех элементов И

Вопросы и задания

1. Имеется схема ИЛИ, у которой есть пять входов и выход. Сколько комбинаций входных сигналов может быть? Во скольких случаях на выходе будет сигнал, и во скольких случаях его не будет?
2. Имеется логическая схема, состоящая из трех элементов ИЛИ, включенных так, как показано на рис. 3.16. Сколько комбинаций сигналов может быть на входе? При каком количестве входных комбинаций на выходе будет сигнал? При каком количестве не будет?
3. Имеется логическая схема, состоящая из трех элементов И (рис. 3.17). Сколько комбинаций сигналов может быть на входе? При каком количестве входных комбинаций на выходе будет сигнал? При каком количестве не будет?
4. По логической схеме рис. 3.16 запишите функцию и постройте комбинаторную диаграмму.
5. По логической схеме рис. 3.17 запишите функцию и постройте комбинаторную диаграмму.

3.5. Программное обеспечение персонального компьютера

Как мы уже говорили, компьютер работает под управлением программ. Вся совокупность программ называется *программным обеспечением* (ПО). Программы, которые поставляются вместе с компьютером и обеспечивают работоспособность компьютерной системы, составляют раздел *системного программного обеспечения*. Основная часть этого раздела содержит программы, входящие в *операционную систему* (ОС). Программы, обеспечивающие решение различного рода прикладных задач, например проектирования, экономических расчетов, управления, статистики, создания анимаций, составляют раздел *прикладного программного обеспечения*. Следует сразу сказать, что эта граница весьма условна.

Современный компьютер немыслим без ОС. Обычно ОС поставляется вместе с компьютером. У всех на слуху их названия: MS DOS, Windows, Unix, Linux, Mac Os и др. Каково назначение ОС? Ее главное назначение — управление ресурсами компьютерной системы: временем центрального процессора, оперативной памятью и файлами на внешних запоминающих устройствах. До создания ОС необходимость в таком управлении также была, но эту проблему решал человек, сидящий непосредственно за пультом ЭВМ и отлаживающий прикладную программу. При этом значительную часть своего времени он тратил именно на управление ресурсами и только оставшееся

время на свою задачу. Конечно, появление ОС позволило значительно поднять уровень эффективности такого управления, в частности, все современные компьютерные системы именно благодаря ОС стали мультизадачными, т. е. при наличии одного процессора в системе работает несколько задач, каждая из которых по специальным алгоритмам получает доступ к ресурсам компьютерной системы, и в конкретный момент выполняется одна из задач. Кроме того, ОС обеспечивают более простой, наглядный *интерфейс* (систему взаимодействия) между человеком и компьютером, упрощают множество действий человека и др. Современные ОС имеют графический интерфейс, что делает работу с ними достаточно простой и удобной.

Как известно, после загрузки ОС Windows на экране возникает так называемый *рабочий стол*. Как и на обычном рабочем столе, на нем могут быть представлены те инструменты, которые наиболее часто используются вами при работе на компьютере. Например, рабочий стол может выглядеть так, как показано на рис. 3.18. Как мы видим, он состоит из ярлыков (пиктограмм) отдельных программ, фоновой картинки (заставки), кнопки **Пуск** и ряда значков в нижней части экрана. После установки Windows, как правило, автоматически по-

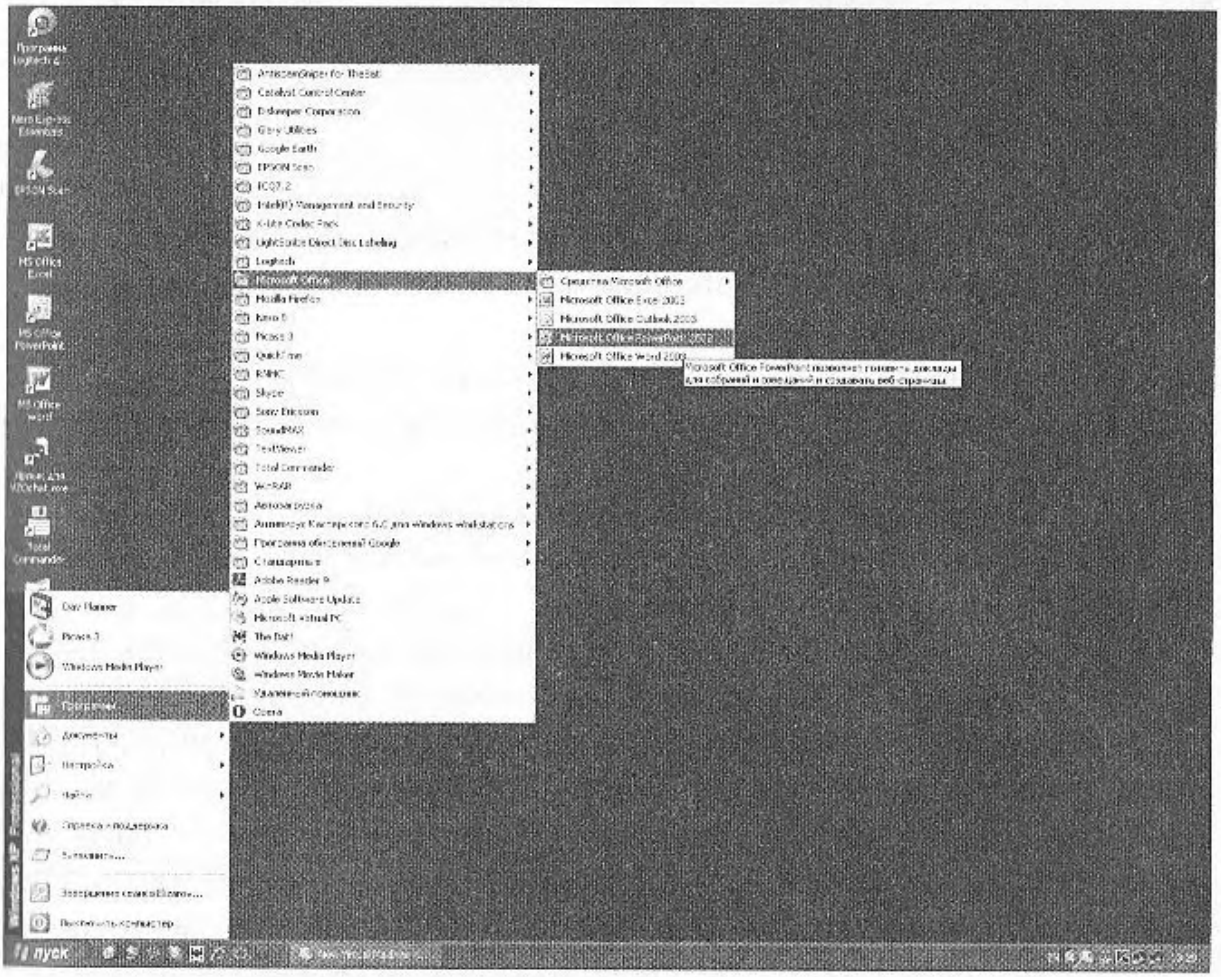


Рис. 3.18. Рабочий стол Windows

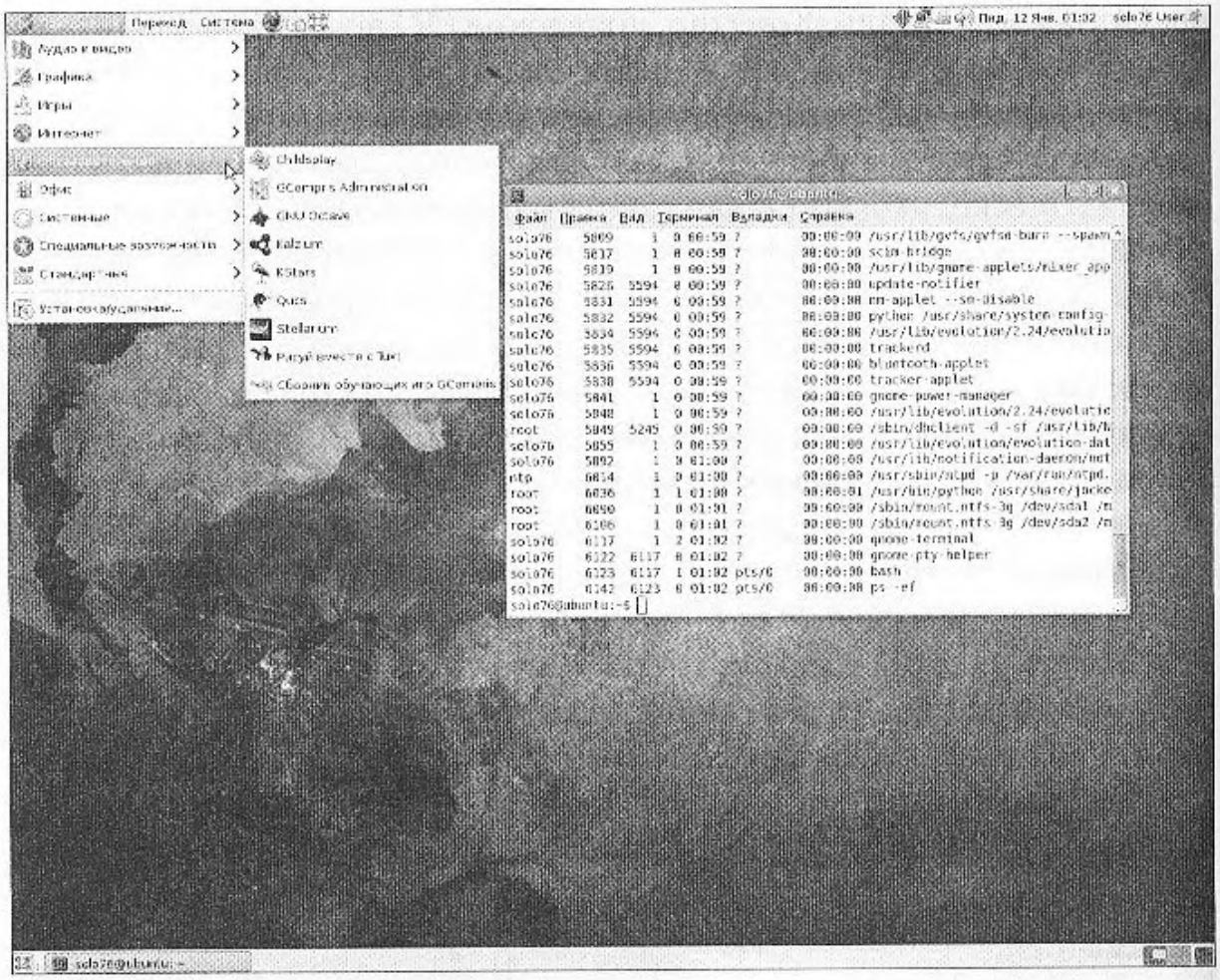


Рис. 3.19. Рабочий стол Ubuntu

являются ярлыки «Мой компьютер», «Сетевое окружение», «Мои документы», «Корзина». Остальные создаются следующим образом:

1. Нажать правую кнопку мыши.
2. В появившемся меню выбрать команду **Создать**.
3. Из предложенной альтернативы «Папка» или «Ярлык» выбрать второе.
4. В появившемся окне нажать кнопку **Обзор**.
5. Выбрать головной модуль программы, ярлык которой требуется вынести на рабочий стол и нажать **ОК**.

При этом следует помнить, что основные программы находятся в папке «Programs Files», которая расположена в корневом каталоге.

Для оформления рабочего стола достаточно нажать правую кнопку мыши и в появившемся меню выбрать **Рабочий стол** (Active Desktop). Используя появившееся меню, можно настроить элемент рабочего стола, определить его вид и т. д. Мелкие значки в нижней части экрана предназначены для быстрого вызова отдельных программ, таких как универсальный мультимедиапроигрыватель, регулятор звука, программа вызова Интернета (браузер) и т. д.

При работе, например, в одной из операционных Unix-систем Ubuntu (http://www.ubuntu.com/getubuntu/download) рабочий стол может выглядеть так, как показано на рис. 3.19.

На рабочем столе также можно разместить ярлыки программ и фоновую картинку. Но так как ядро Unix является свободно распространяемым, существует множество ОС на основе этого ядра. Поэтому названия кнопок и пункты меню рабочего стола могут меняться в зависимости от установленной ОС.

Работа с файлами может проводиться как через графический интерфейс (например, при работе с файловым менеджером), так и с помощью командной строки. На рис. 3.20 представлен *файловый менеджер* Frigate (http://www.frigate3.com/download/frigate3_pro.exe). Практически все операции с файлами являются интуитивно понятными и не требуют каких-либо особых знаний.

В случае работы в режиме командной строки все действия выполняются с помощью специальных команд (для Windows и Unix-систем языки, на которых пишутся команды, различаются).

И тем не менее главное назначение ОС — управление ресурсами компьютера. Операционные системы осуществляют контроль за работоспособностью аппаратной части компьютерной системы, содержат

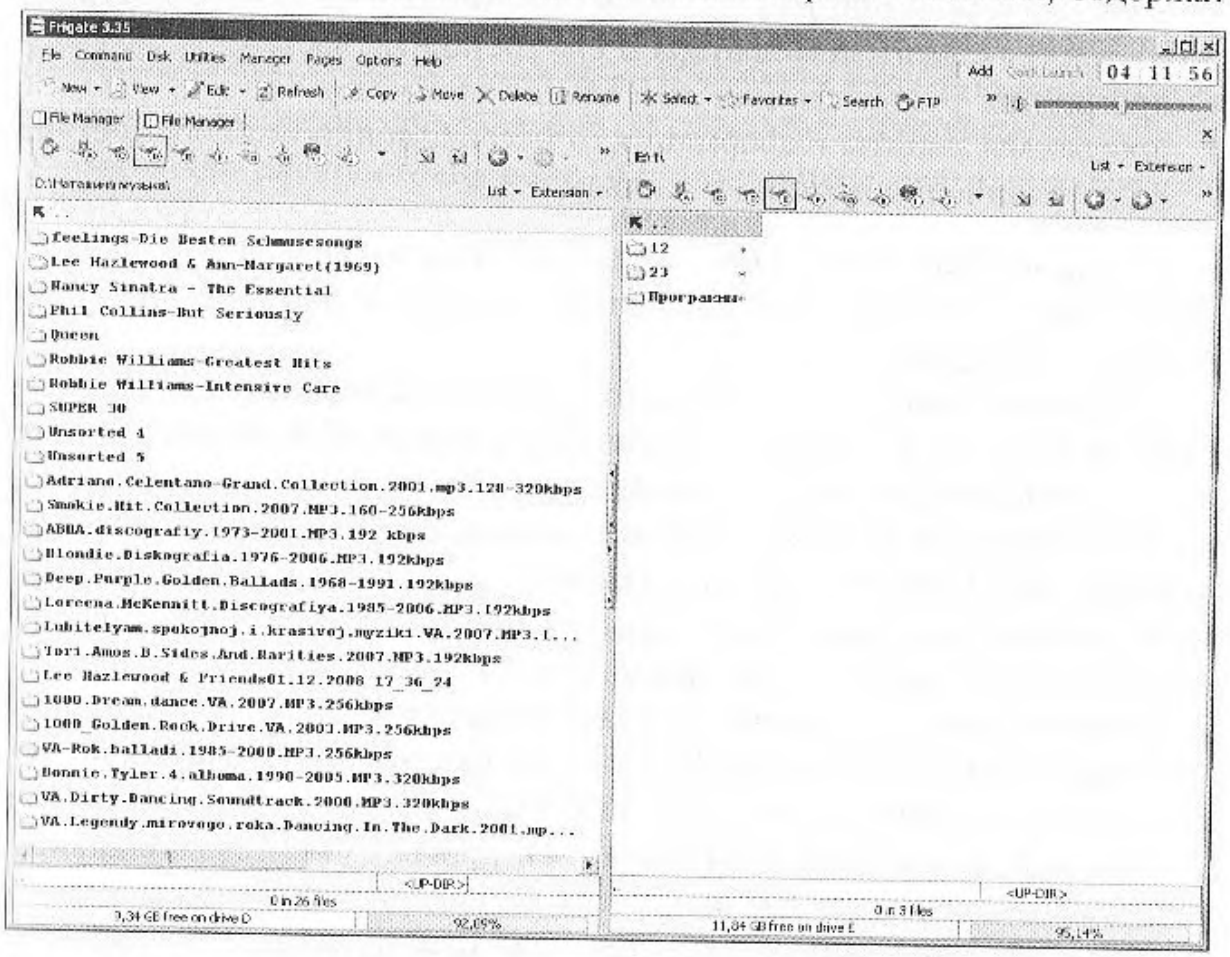


Рис. 3.20. Файловый менеджер Frigate

средства диагностики и поиска неисправностей, позволяют оптимальным образом использовать ресурсы, обеспечивают работу компьютера в различных сетях (локальных и глобальных), дают возможность подключать к компьютеру самые разнообразные устройства, включая соответствующие программы — *драйверы*, учитывают дату и время за счет *системного таймера*, содержат множество программ, обеспечивающих выполнение элементарных системных действий. В Windows эти программы содержатся в специальных динамически загружаемых библиотеках (DLL). К числу системных программ относят *программы-оболочки*, большое количество вспомогательных, сервисных программ — *утилит*, в частности:

- программы резервирования, обеспечивающие надежность хранения информации;
- антивирусные программы, позволяющие бороться с компьютерными вирусами;
- архиваторы, позволяющие хранить информацию в упакованном формате, используя меньшее количество памяти;
- программы-перекодировщики, обеспечивающие, например, русификацию клавиатуры;
- дагностические программы, позволяющие осуществлять контроль состояния самого компьютера и его составных частей;
- программы ускорения или замедления процессов;
- программы защиты информации от несанкционированного доступа и разграничения доступа к данным и т. д.

Очень важной частью ОС является *командный интерпретатор*. Именно он переводит язык привычных команд на язык «железа», понятный аппаратуре.

Мир прикладного ПО настолько огромен и многообразен, что описать его вкратце достаточно трудно. Прикладное ПО можно условно разделить на группы программ общего и специального назначения.

К *программам общего назначения* можно отнести приложения (на примере MS Office) — это текстовый редактор Word, электронные таблицы Excel, мастер создания презентаций PowerPoint, СУБД Access, графические редакторы (например, Paint).

Широко известны такие прикладные пакеты, как Photoshop, применяемый для обработки графической информации; CorelDRAW, используемый в издательском деле; СУБД (например, Oracle); программы управления правовыми БД («Гарант»); программы — персональные информационные менеджеры (Lotus Organizer); программы планирования, необходимые, в частности, для планирования выполнения проекта (MS Project); программы распознавания символов, исполь-

зующиеся при сканировании (FineReader), программы — переводчики с иностранных языков (Lingvo). Все эти программы имеют удобный графический интерфейс, достаточно просты в освоении и широко используются. К этому же классу можно отнести системы программирования, позволяющие создавать новые программы и включающие в себя входной язык, транслятор (переводчик на язык компьютера) и библиотеки программ, облегчающих процесс программирования.

Специализированное программное прикладное обеспечение менее известно широкому кругу компьютерных пользователей, зато его активно применяют в тех профессиональных сферах, для которых оно и создано. Например, в системах автоматизированного проектирования эффективно применяется программа AutoCAD. Широко известны бухгалтерские программы 1С, «Турбо Бухгалтер» и т.д.

Наряду с коммерческими программными продуктами широкое распространение получило так называемое *«свободное» программное обеспечение*. Дистрибутивы (программы, готовые для установки на компьютере — инсталляции) размещены в Интернете для широкого круга пользователей. Большинство разработчиков свободного ПО позиционирует свои продуты как «free for non commercial use» — свободные для некоммерческого использования. Это означает, что такое ПО может быть свободно использовано частными лицами и образовательными учреждениями, но не может использоваться кем-либо для изготовления коммерческих продуктов.

Свободно распространяемое ПО, популярное для использования в образовательных целях, приведено в табл. 3.12.

Таблица 3.12

Название ПО	Назначение	Дистрибутив
ALT Linux Edubuntu Debian MOPSLinux	Операционные системы (для серверов и рабочих станций)	http://linux.armd.ru http://ubuntu.ru/ http://www.debian.org/ http://www.linux.org/
OpenOffice	Офисный пакет (текстовый редактор Writer, редактор математических формул Math, электронная таблица Calc, система управления базами данных Base, редактор презентаций Impress)	http://ru.openoffice.org/
Scratch (аналог Logo)	Визуальная среда программирования	http://scratch.mit.edu/

Окончание табл. 3.12

Название ПО	Назначение	Дистрибутив
Lazarus (аналог Pascal)	Визуальная среда программирования	http://www.lazarus.freepascal.org/
Eclipse	То же	http://www.eclipse.org/
NetBeans	»	http://www.netbeans.org/
Mozilla Firefox	Web-браузер	http://www.mozilla.org/ru/firefox/
Mozilla Thunderbird	Клиент электронной почты	http://www.mozilla.org/ru/thunderbird/
KompoZer	HTML-редактор	http://www.kompozer.net/
Gimp	Редактор растровой графики	http://www.gimp.org/
Inkscape	Редактор векторной графики	http://www.inkscape.org/
QCad	Профессиональная система черчения	http://www.qcad.org/
Scribus	Настольная издательская система	http://www.scribus.net/
Audacity	Редактор звуковых файлов	http://audacity.sourceforge.net/
Avidemux	Видеоредактор	http://www.avidemux.org/
Blender	Редактор трехмерной графики	http://www.blender.org/
Skype	Видеосвязь, служба мгновенных сообщений	http://www.skype.com
MySQL	Система управления базами данных	http://www.mysql.com/
Google Apps	Software as a Service (ПО как сервис)	http://www.google.com/a/help/intl/ru/index.html

Вопросы и задания

1. В чем состоит основное назначение операционной системы?
2. Какие программы называются утилитами?
3. Сравните программный интерфейс двух разных ОС (цв. вкл., рис. 1, 2).
4. Какие программы называют драйверами?
5. Назовите программы, входящие в состав MS Office. Какие аналоги можно использовать среди свободного ПО?

3.6. Защита информации

Для более компактного представления содержимого файлов при записи на внешние носители (съемные), пересылке по электронной почте, при длительном хранении в целях экономии дискового пространства их целесообразно упаковывать, т.е. *архивировать*. Для нормальной работы с архивным файлом его предварительно нужно распаковать — *разархивировать*. Для этих целей существуют специальные программы — **архиваторы**. Наиболее популярной для Windows является программа WinRAR (свободную версию этой программы можно скачать по ссылке http://www.rarlab.com/download.htm). Рассмотрим принцип

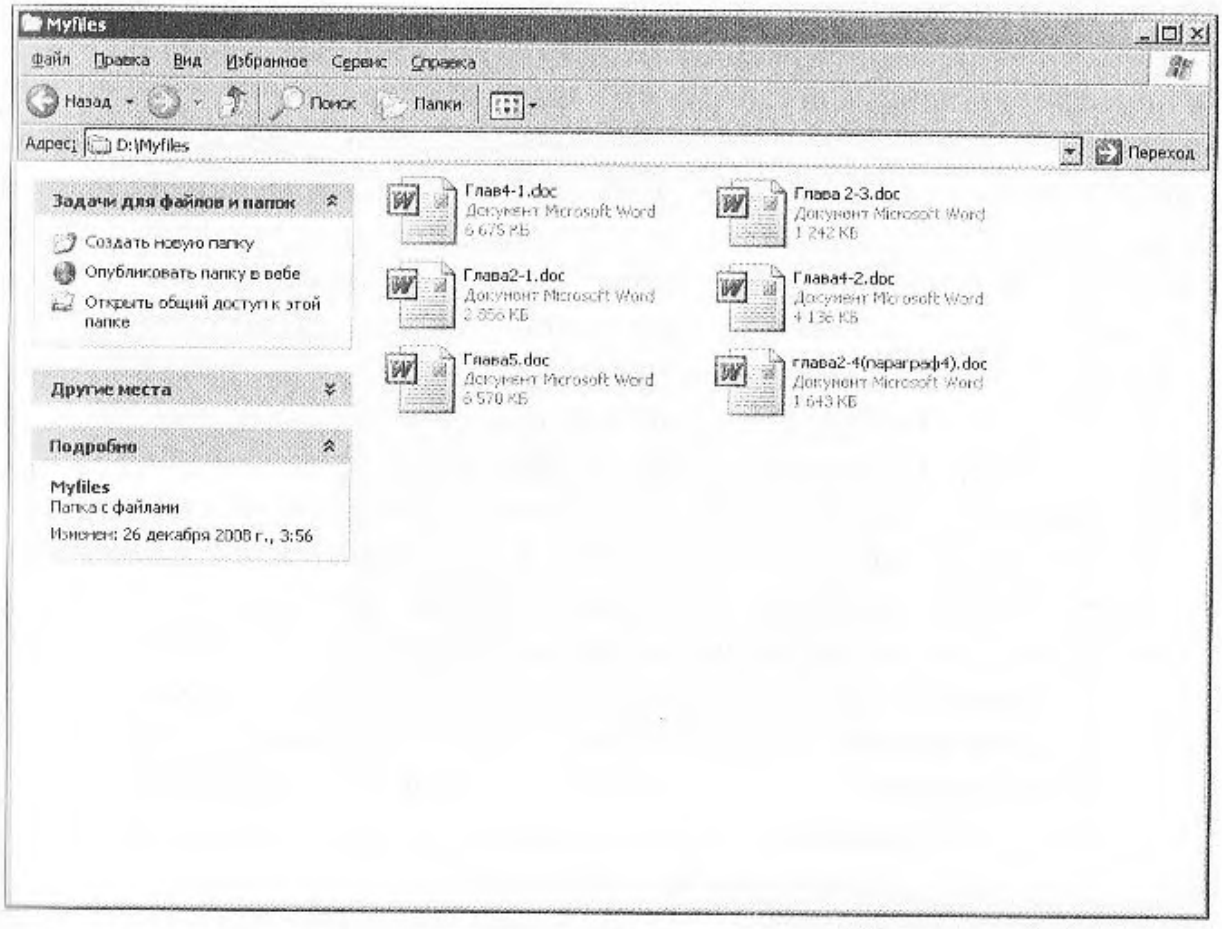


Рис. 3.21. Содержимое директории Myfiles

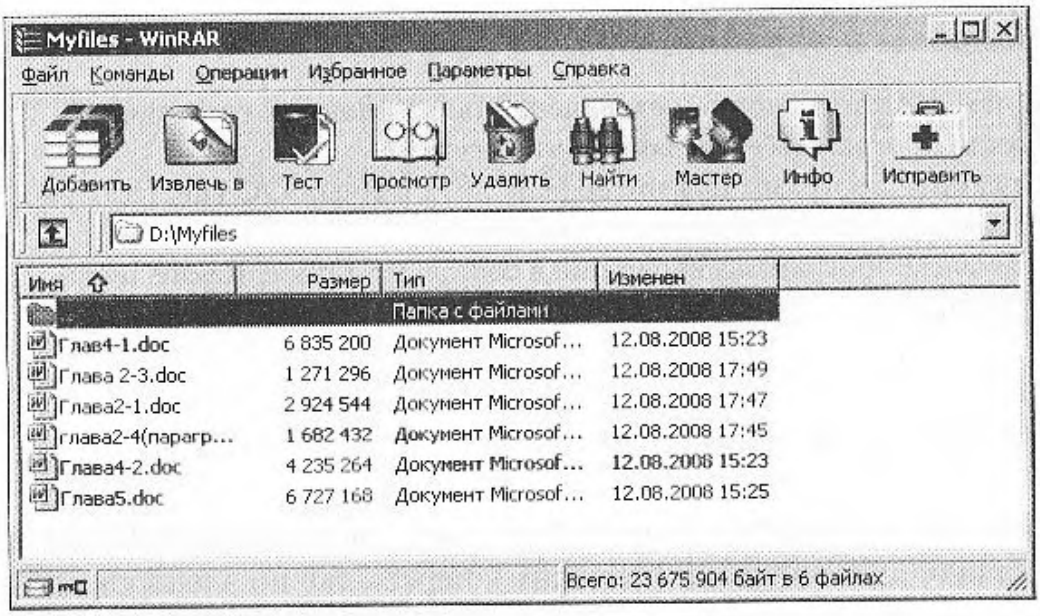


Рис. 3.22. Интерфейс WinRar

ее работы. Предположим, у вас есть папка «Myfiles», содержащая ряд файлов — текстовых документов (рис. 3.21). Эти документы нужно поместить на одну флеш-память. Суммарный размер этих файлов составляет 1,91 Мбайт, а на вашем носителе информации осталось меньше места. Попробуем заархивировать эту папку, для чего нажмем **Пуск**, **Программы**. Если на компьютере установлен WinRAR, то он появится в списке программ. Запустим его и выберем в появившемся окне нужную папку (рис. 3.22). Все файлы, которые необходимо поместить в архив, нужно предварительно выделить. Для этого можно, удерживая кнопку **Ctrl**, щелкнуть левой кнопкой мыши на имена файлов. Если нужно выделить все файлы папки, как в нашем случае, следует в меню **Файл** (File) выбрать команду **Выделить все** (Select all). После этого все имена будут изображены на синем фоне. Затем нажмем на левую клавишу в верхней части окна архиватора **Добавить** (Add). В результате появится новое окно, в котором нужно указать метод архивирования (сжатия), формат архивирования (WIN или ZIP), имя архивного файла и параметры архивирования (рис. 3.23).

Имя и местоположение архивного файла выберем по умолчанию (в данном случае это будет имя папки). Метод архивирования лучше использовать тот, который предлагается по умолчанию — «Обычный». Методы архивирования отличаются тем, что в разных методах используются разные параметры, в одном минимально короткое время, в другом — максимальная степень сжатия, независимо от времени и т.д. Итак, выполнив **Добавить**, в появившемся окошке увидим, как происходит процесс архивирования (рис. 3.24).

В результате в папке появился файл «Myfiles.rar» размером 549 Кбайт, который и является архивом всех файлов данной папки.

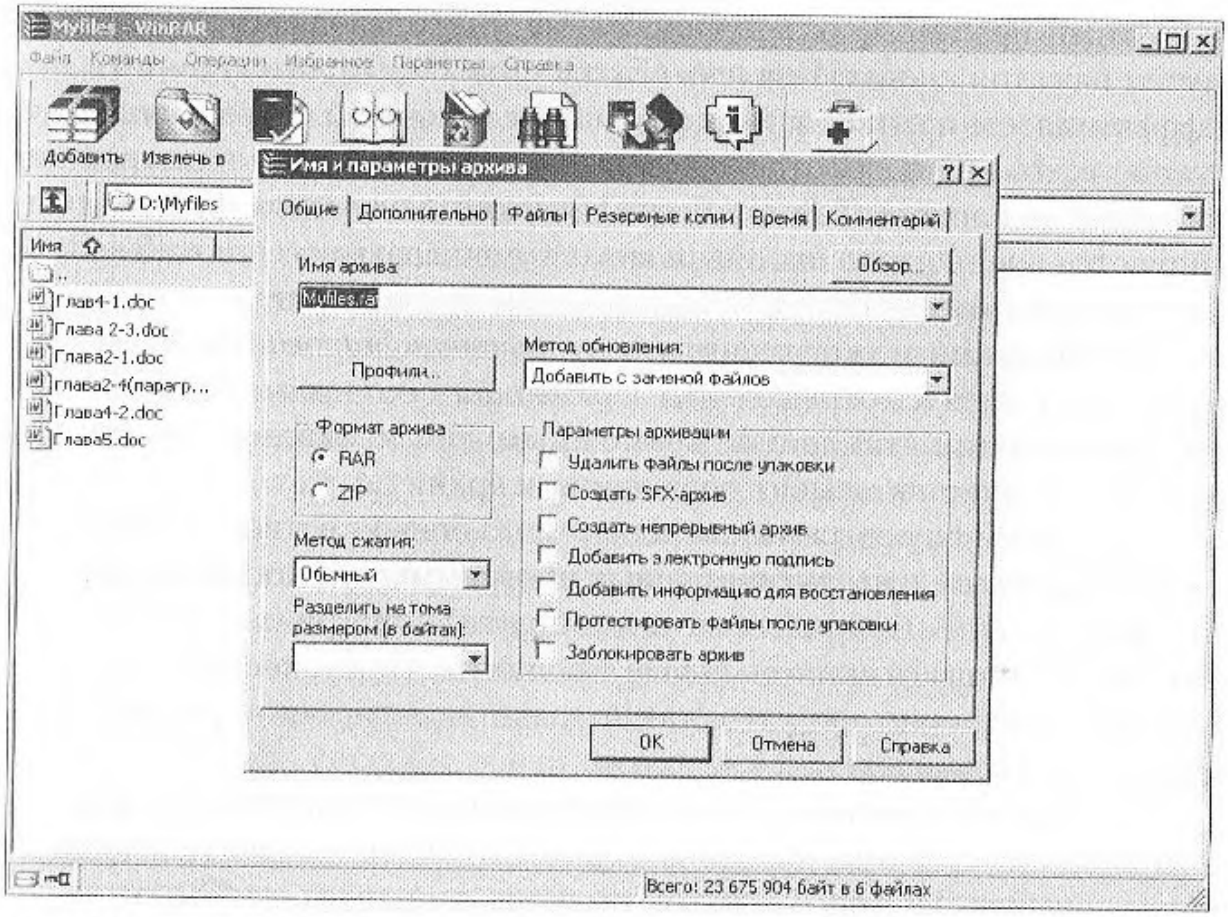


Рис. 3.23. Задание параметров архивирования

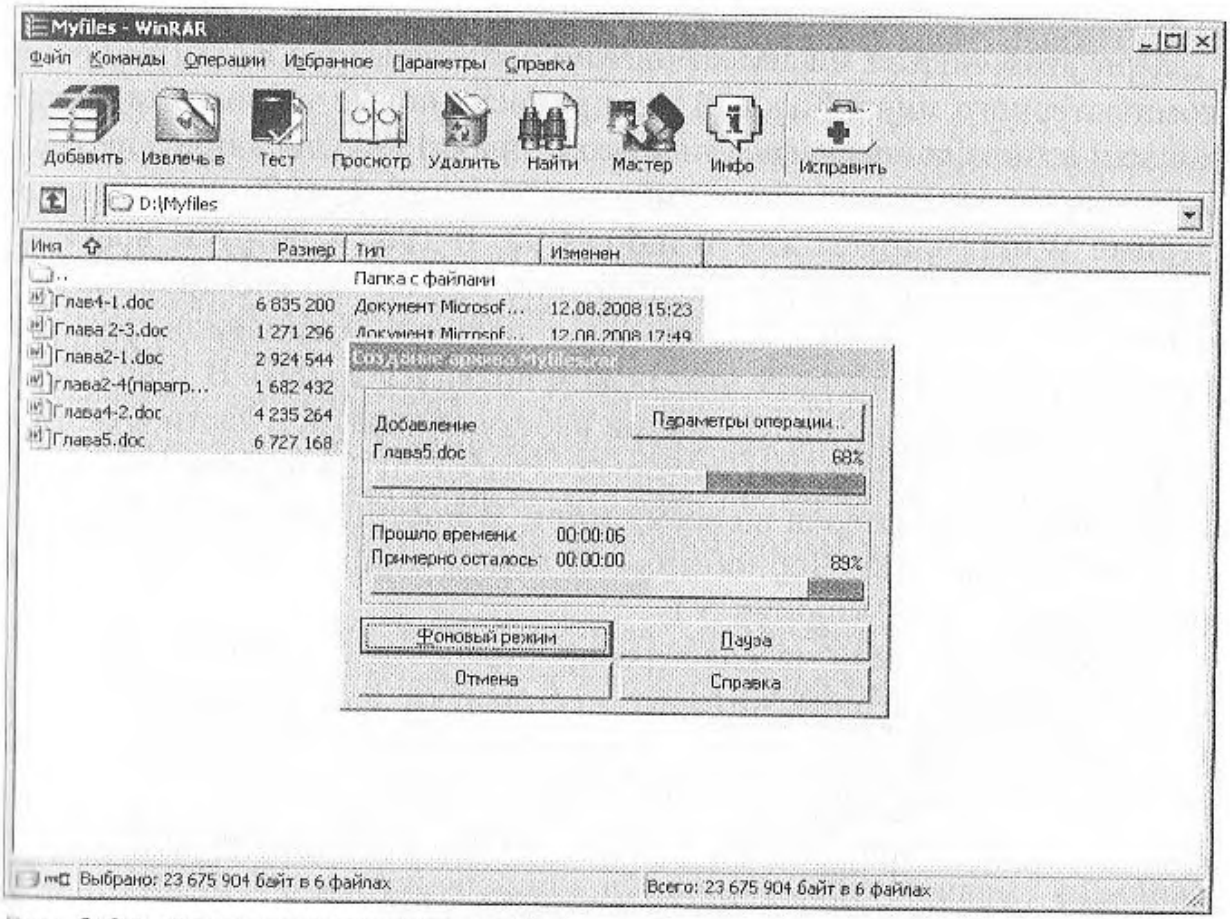


Рис. 3.24. Процесс создания архива

Обратный процесс — разархивирование — производится аналогично, путем нажатия кнопки **Извлечь** (Extract to). Необходимо помнить, что эффективность архивирования различных типов файлов различна. Так, лучше всего архивируются текстовые файлы, файлы — электронные таблицы, неплохо — БД, а вот графические и звуковые файлы архивируются очень плохо, их объем может уменьшаться всего на 5—10 %. Это нужно учитывать при использовании архиваторов.

Любой архив можно защитить от *несанкционированного доступа*. Для этого предусмотрен *режим архивации с паролем*. В результате раскрыть архив становится возможным при правильно введенном пароле на запрос в момент обращения к архиву.

Среди набора программ, используемого большинством пользователей ПК каждый день, **антивирусные программы** традиционно занимают особое место. Современные антивирусные программы представляют собой многофункциональные продукты, сочетающие в себе как профилактические средства, так и средства лечения вирусов и восстановления данных.

Требования к антивирусным программам достаточно противоречивы. С одной стороны, пользователи хотят иметь надежную, мощную *антивирусную защиту*. С другой стороны, они хотят, чтобы эта защита не требовала от пользователя много времени и сил. И это вполне естественные требования.

При этом нельзя ни на мгновение отставать от общего развития компьютерного мира. Каждый год приносит новые технологии, в том числе и в мире *компьютерных вирусов*. Так, в 1995 г. появился первый макровирус, заражающий документы MS Word. В 1996 г. появились первые Win32-вирусы для Windows 95. В 1997 г. вирусы впервые стали использовать для распространения сообщения электронной почты и появились первые вирусы, работающие в защищенном режиме процессоров Intel (впервые этот режим появился в i286). В 1998 г. был создан первый вирус, нарушающий работу аппаратной части компьютеров. Это был Win95.CIH, который «сработал» 26 апреля 1999 г. на миллионах компьютеров по всему миру. В России этот вирус стал известен под именем «Чернобыль». В самом конце 1998 г. появился первый вирус для Windows NT.

В 1999 г. получили массовое распространение e-mail-черви (вирусные программы-черви), которые используют для распространения сообщения электронной почты. Эпидемия вируса Win95.Spanska.10000 («Happy99») началась 1 января 1999 г. и продолжается до сих пор. Другой e-mail-червь Melissa в марте 1999 г. парализовал работу нескольких тысяч почтовых серверов в Европе и Америке. По масштабу мартовскую эпидемию червя Melissa можно сравнить с легендарным

«червем Морриса», который в ноябре 1988 г. парализовал работу нескольких крупных компьютерных сетей в Америке.

Также в 1999 г. стали очень популярны троянские программы (троянские системы семейств Back Orifice, NetBus, Trojan Stealth), дающие удаленный доступ к инфицированному компьютеру через Интернет и позволяющие воровать информацию, например пароли.

Все эти новинки заставляют постоянно совершенствовать антивирусные программы. Пользователю важно лишь не забывать об угрозе компьютерных вирусов и принимать для защиты от них меры, не требующие в принципе больших усилий или специальных знаний. Достаточно проводить регулярное резервное копирование важных данных и пользоваться современными антивирусными программами.

Способы противодействия компьютерным вирусам можно разделить на несколько групп: профилактика вирусного заражения и уменьшение предполагаемого ущерба от такого заражения; методика использования антивирусных программ, в том числе обезвреживание и удаление известного вируса; способы обнаружения и удаления неизвестного вируса.

Однако сразу хотелось бы отметить, что нет антивирусов, гарантирующих 100%-ю защиту от вирусов. Таких систем не существует, поскольку на любой алгоритм антивируса всегда можно предложить контралгоритм вируса, невидимого для этого антивируса (обратное, к счастью, тоже верно: на любой алгоритм вируса всегда можно создать антивирус).

Следует также обратить внимание на несколько терминов, применяемых при обсуждении антивирусных программ:

- «ложное срабатывание» (false positive) — определение вируса в незараженном объекте (файле, секторе или системной памяти). Обратный термин — false negative;
- «сканирование по запросу» (on-demand) — поиск вирусов по запросу пользователя. В этом режиме антивирусная программа неактивна до тех пор, пока не будет вызвана пользователем из командной строки, командного файла или программы-расписания (system scheduler);
- «сканирование на лету» (real-time, on-the-fly) — постоянная проверка на вирусы объектов, к которым происходит обращение (запуск, открытие, создание и т. п.). В этом режиме антивирус постоянно активен, он присутствует в памяти постоянно (резидентно) и проверяет объекты без запроса пользователя.

Принципы диагностики компьютера на основе антивирусных программ приведены ниже.

1. Следите за тем, чтобы антивирусные программы, используемые для проверки, были самых последних версий. Если к программам поставляются обновления, то проверьте их на «свежесть».
2. «Национальность» антивирусов в большинстве случаев не имеет значения, поскольку на сегодняшний день процесс эмиграции вируса в другие страны и иммиграции антивирусных программ ограничивается только скоростью Интернета, поэтому как вирусы, так и антивирусы не признают границ. Если на компьютере обнаружен вирус, то самое главное — не паникуйте. Если вирус обнаружен в каком-то из новых файлов и еще не проник в систему, то нет причин для беспокойства: удалите вирус любимой антивирусной программой и спокойно работайте дальше.
3. В случае обнаружения файлового вируса, если компьютер подключен к сети, отключите его от локальной сети и проинформируйте системного администратора. Если вирус еще не проник в сеть, это защитит сервер и другие рабочие станции от проникновения вируса. Если же вирус уже поразил сервер, то отключение от сети не позволит ему вновь проникнуть на компьютер после его лечения. Подключение к сети возможно лишь после того, как будут вылечены все серверы и рабочие станции.
4. Если обнаружен файловый или загрузочный вирус, убедитесь в том, что вирус либо перезидентный, либо резидентная часть вируса обезврежена: при запуске некоторые (но не все) антивирусы автоматически обезвреживают резидентные вирусы в памяти. Удаление вируса из памяти необходимо для того, чтобы остановить его распространение.

С помощью антивирусной программы можно восстановить зараженные файлы и затем проверить их работоспособность. Перед лечением или одновременно с ним — создать резервные копии зараженных файлов и распечатать или сохранить где-либо список зараженных файлов (log-файл антивируса). Это необходимо для того, чтобы восстановить файлы, если лечение окажется неуспешным из-за ошибки в лечащем модуле антивируса либо по причине неспособности антивируса лечить данный вирус. В этом случае придется прибегнуть к помощи какого-либо другого антивируса.

Одним из основных методов борьбы с вирусами является, как и в медицине, *своевременная профилактика*. Компьютерная профилактика предполагает соблюдение небольшого числа правил, которое позволяет значительно снизить вероятность заражения вирусом и потери каких-либо данных.

Для того чтобы определить основные правила компьютерной гигиены, необходимо выяснить основные пути проникновения вируса в компьютер и компьютерные сети.

Основным источником вирусов на сегодняшний день является глобальная сеть Интернет. Наибольшее число заражений вирусом происходит при обмене письмами в форматах Word. Пользователь зараженного макровирусом редактора Word, сам того не подозревая, рассылает зараженные письма адресатам, которые в свою очередь отправляют новые зараженные письма и т.д.

Предположим, что пользователь ведет переписку с пятью адресатами, каждый из которых также переписывается с пятью адресатами. После посылки зараженного письма все пять компьютеров, получившие его, оказываются зараженными. Затем с каждого вновь зараженного компьютера отправляется еще пять писем. Одно уходит назад на уже зараженный компьютер, а четыре — новым адресатам.

Таким образом, на втором уровне рассылки заражено уже 1 + 5 + + 20 = 26 компьютеров. Если адресаты сети обмениваются письмами один раз в день, то к концу рабочей недели (за пять дней) зараженными окажутся как минимум 1 + 5 + 20 + 80 + 320 = 426 компьютеров. Нетрудно подсчитать, что за 10 дней зараженными оказываются более 100 000 компьютеров! Причем каждый день их количество будет учетверяться.

Электронные конференции также служат одним из основных источников распространения вирусов. Практически каждую неделю приходит сообщение о том, что какой-либо пользователь заразил свой компьютер вирусом, который был снят с какой-либо электронной конференции. При этом часто зараженные файлы «закладываются» автором вируса и рассылаются по нескольким конференциям одновременно, а эти файлы маскируются под новые версии какого-либо ПО (иногда под новые версии антивирусов).

Главный путь быстрого заражения — локальные сети. Если не принимать необходимых мер защиты, то зараженная рабочая станция при входе в сеть заражает один или несколько служебных файлов через компьютерную сеть.

Нелегальные копии ПО, как это было всегда, являются одной из основных зон риска. Часто пиратские копии на компакт-дисках содержат файлы, зараженные самыми разнообразными типами вирусов.

Файлы, с которыми ведется работа, необходимо периодически сохранять на внешнем носителе. Такие резервные копии носят название *backup-копий*. Затраты на копирование файлов, содержащих исходные тексты программ, БД, документацию, значительно меньше затрат на

восстановление этих файлов при проявлении вирусом агрессивных свойств или сбое компьютера.

? Вопросы и задания

1. Что такое архиватор?
2. Что такое антивирусная программа?
3. Опишите технологию использования антивирусных программ.
4. Откуда берутся вирусы?
5. Перечислите меры профилактики вирусов

П Проектное задание

Подготовьте инструкцию «Профилактика вирусов ПК» на основе антивирусной программы, используемой вами.

ГЛАВА 4

ТЕХНОЛОГИИ СОЗДАНИЯ И ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИОННЫХ ОБЪЕКТОВ

4.1. Технология обработки текстовой информации

Современные компьютерные инструменты обработки информации подразделяются на три составляющие: обработки текста, числовой информации, рисования и звукового оформления. Программное обеспечение, позволяющее пользователю активно использовать в самостоятельной компьютерной деятельности все эти возможности комплексно, можно определить как *редакторские программы*. Поэтому основная задача — узнать их общие свойства и научиться пользоваться в дальнейшем любой редакторской средой.

Программные среды, имеющие единые правила взаимодействия между собой, называются *интегрированными*. Среда Office включает в себя такие возможности. Например, MS Windows в списке программ **Стандартные** предлагает группу простейших редакторов: для обработки текста — WordPad, «Блокнот»; для обработки числовых данных — «Калькулятор»; для обработки графики — Paint. Все эти программы являются редакторами и имеют одинаковые инструменты оформления файла, настроек страницы (если она предусмотрена), управления набором символов в поле строки.

Существуют компьютерные средства, в которых можно использовать все указанные возможности комплексно. Такие программные среды называются *мультимедийными*. Например, в компьютерных презентациях используется текст с гиперссылками, числовые таблицы, рисунки и видео, звуковое оформление, а также возможности анимации и управления объектами презентации — слайдами.

Для того чтобы обработать текст на компьютере, требуется специальная программа, с помощью которой можно оформить текстовый документ. **Текстом** для ПК можно называть набор данных, состоящий из символов языка (буквы, цифры, знаки). Файл, содержащий инфор-

мацию текстового типа, называется *текстовым*. Для того чтобы компьютерная программа распознала тип данных в файле, нужно в полном имени файла указать его тип. Например, программам-редакторам соответствуют следующие типы файлов:

- WordPad – RTF;
- Word – DOC;
- «Блокнот» – TXT.

Часто программа-редактор приписывает тип файла к указанному названию автоматически. Название файла задается так, чтобы было легко в дальнейшем по нему ориентироваться.

Программа для обработки текста на компьютере называется **текстовым редактором**. Программы, обладающие большими возможностями для обработки текста, включая оформление их иллюстрациями, таблицами, разнообразными видами шрифтов, в том числе автоматическую проверку орфографии, сборку оглавления, списков литературы, иллюстраций, нумерацию страниц, называются **текстовыми процессорами**.

Текстовые редакторы предлагают диалог в форме открывающихся окон ответов и подсказок. Используя пиктограммы меню управления окном, его можно распахнуть, свернуть и закрыть.

Все текстовые редакторы имеют в своем рабочем окне *строку-меню* и *строку-справку*, содержащие меню инструментов для обработки текста и справку о текстовом файле.

Текстовый редактор позволяет выполнять с текстом следующие действия:

- набор текста (быстрый способ набора – десятипальцевая печать);
- исправление и дополнение текста с помощью специальных клавиш и меню инструментов;
- сохранение текста на магнитном диске с использованием правил работы с файлами и библиотекой файлов;
- вывод содержимого файла на устройства вывода: просмотр на экране дисплея, печатать на бумаге, красивое расположение на ней текста.

Текст вводится с клавиатуры, а выводится либо на экран, либо на печатающее устройство, либо запоминается в виде текстового файла на внешнем носителе.

Какие существуют новые способы ввода текста, кроме ручного набора на клавиатуре? Выясните, как помогают нам вводить текст сканер, компьютерное перо. Почему для формирования текстового файла из графического образа текста, полученного с помощью сканера или пера, необходима специальная программа – распознаватель текста (например, FineReader)?



Рис. 4.1. Схема постановки рук при 10-пальцевой печати

Компьютеризация машинописных работ имеет большие преимущества. Удобство работы в текстовом редакторе в отличие от пишущей машинки состоит в том, что, напечатав неправильное слово, лист не нужно перепечатывать — его легко исправить, можно добавлять фрагменты текста в любом месте документа, использовать разные способы изображения букв — шрифты, цвета и т. д. Однако содержимое текстового файла, если оно является документом (письмом, справкой свидетельством и т. п.), становится реальным только тогда, когда оно выведено на бумагу. Тогда на текстовом файле можно поставить печать, расписаться, и он приобретает силу документа. В настоящее время многие страны вводят в оборот понятия «цифровая подпись» и «электронный документооборот». Это означает, что электронный документ с подписью в ряде случаев приобретает юридическую силу.

Для использования текстового редактора необходимо уметь быстро набирать текст на клавиатуре (рис. 4.1), исправлять текст по правилам, предложенным текстовым редактором (эти правила можно посмотреть во вкладке **Помощь — Help**), сохранять текст на магнитном диске и печатать готовый документ на бумаге.

Клавиши простого редактирования текста и команды редактирования текстовых фрагментов предлагаются каждым текстовым редактором (табл. 4.1).

Проведите эксперимент, набирая символы клавиатуры совместно с клавишей **Shift**, и определите: как установить запятую, значок номера, знак вопроса, кавычки, скобки, заглавную букву и другие символы.

Выясните, как перейти в режим латинских букв и обратно в режим кириллицы (русских букв).

Таблица 4.1

Клавиша	Операция
Пробел	Вставка пробела между символами
Удаление символа	
Delete	Удаление символа — «ластик» в позиции курсора
Back space (←)	Удаление символа слева от курсора («забой» справа)
Вставка пропущенного символа	
Стрелки или мышь, клавиша с символом	Подведение курсора к нужному месту, вставка нужного символа
Insert (режим «Вставка»)	Вставка набираемых символов в позицию курсора, остальные — сдвиг вправо
Замена имеющихся символов	
Insert (режим «Замена»)	Расположение набираемых символов поверх имеющихся, замена их собой
Выделение группы символов	
Проведение мышью с нажатой левой клавишей по группе символов	Выделение отмеченной группы символов фоном
Перенос выделенной группы символов	
Подведение курсора к нужному месту, выделение группы символов, Ctrl + Insert, Shift + Insert	Запись выделенной группы символов в специальную память — буфер; выделенная группа символов вставится, начиная с позиции курсора, сдвинув вправо находящиеся там ранее символы
Переход на новую строку	
Shift + Enter	Принудительный переход на строчку без закрытия абзаца
Enter	Закрытие абзаца

Абзац — это строительный «кубик» текста. Текстовые редакторы выполняют автоматически переход на новую строчку текста при достижении границы полей на странице. Нажатие клавиши **Enter** закрывает абзац и начинает новый с отступа («красной» строки). Настройка абзаца называется в редакторе *форматированием*. Каждый абзац можно форматировать по-своему. Текстовый редактор имеет стандартные настройки, заложенные в программе.

Познакомьтесь со стандартными настройками и начните работу, не меняя их. Настроить параметры форматирования позволяет команда меню **Абзац**. Чаще всего она заложена в разделе **Формат**.

После того, как текст набран, его нужно сохранить на диске. Команда **Сохранить** (Save) в меню текстового редактора выполняет эту операцию, сохраняя файл под прежним именем (если оно было) или присваивая свое имя. Можно сохранить файл как копию под другим именем. Для этого потребуется команда **Сохранить как** (Save as). Сохранять документ можно неоднократно после каждого изменения текста в нем или в конце работы.

Для создания нового файла в текстовом редакторе необходимо воспользоваться командой **Создать**, и тогда появится «чистый лист» экрана для работы (табл. 4.2).

Таблица 4.2

Команда меню «Файл» в программах-редакторах	Описание
Создать	Создает текстовый файл в оперативной памяти и показывает на экране монитора
Открыть	Пересылает текстовый файл с магнитного диска в оперативную память и показывает на экране монитора
Сохранить	Записывает текстовый файл на магнитный диск по указанному пути, оставляя его на экране монитора. При повторном сохранении содержимое файла будет обновляться под старым именем
Сохранить как	Сохраняет текстовый файл под новым именем

Окончание табл. 4.2

Команда меню «Файл» в программах-редакторах	Описание
Выйти	Закрывает работу с программой, выгружая из оперативной памяти все текстовые файлы, с которыми производилась работа. При этом предлагается сохранить изменения в этих файлах на магнитном диске

Начиная работать с текстом на компьютере, например с текстом реферата, следует заранее построить модель оформления текста.

Одним из важнейших параметров модели оформления текста является **настройка страницы**. Она предполагает положение страницы — книжное или альбомное, поля на странице с учетом отступа на переплет. При этом программа автоматически вынесет отступ слева или справа в зависимости от четности страницы. На странице текст строится из абзацев. Это ключевой инструмент моделирования текста. Необходимо заранее выяснить, как будет оформлен абзац (установить шрифт, поля, отступ первой строки абзаца, выравнивание по краям, межстрочное расстояние) — выбрать его модель. Например, в модели текста на рис. 4.2 четыре абзаца. Каждый абзац имеет свои параметры — положение на странице (отступы), форматирование по центру, а также межабзацное расстояние. Модель текста на странице хорошо просматривается в режиме просмотра текста в масштабе 50 %.

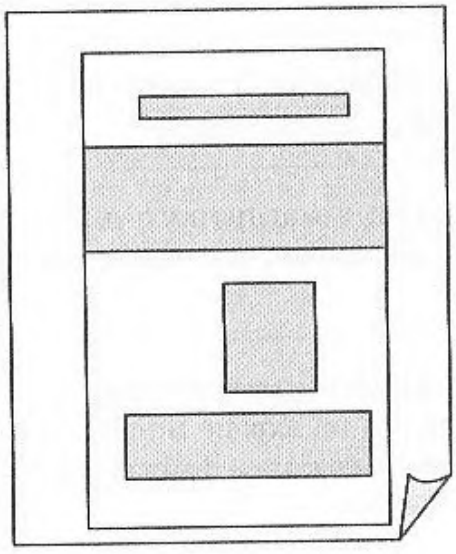


Рис. 4.2. Пример модели расположения текста

Носителем всех настроек абзаца является символ конца абзаца, размещаемый автоматически в конце последней строки абзаца. Возможно переносить настройки выбранного абзаца на другие, для чего в редакторе имеется инструмент копирования настроек, например, в Word это инструмент «копировать формат».

Вопросы и задания

1. Самостоятельно исследуйте текстовые редакторы (например, WordPad и «Блокнот») и выясните, какие инструменты работы с текстом они предлагают, в чем сходство и в чем

отличие их меню. Какой из них позволяет удобнее обрабатывать текст, т.е. имеет следующие возможности:

- автоматический переход на новую строчку при достижении границы текста;
- автоматический отступ «красной» строки;
- возможность использования различных шрифтов;
- возможность изменения размера букв;
- установка в текст картинки;
- установка в тексте звукового сопровождения.

2. Исследуйте меню редакторов с цветной вклейкой, рис. 3, 4 по инструментальным возможностям, предложенным в табл. 4.3.

Таблица 4.3

Команда меню	Описание
Файл, макет страницы (параметры страницы)	Определяет границы текста (верхнее, нижнее, правое и левое поле) на странице, при достижении которых программа автоматически переводит текст на следующую строку, а также устанавливает «книжный» или «альбомный» вид страницы
Вид	Раскрывает или прячет панели инструментов на экране
Вставка	Вставляет в текст объекты: рисунки, формулы, звуки
Формат, шрифт	Изменяет вид, способ изображения и размер букв, выбирая их из библиотеки шрифтов, способов изображения (жирный, наклонный, подчеркивание) и списка размеров
Формат, абзац	Настраивает (форматирует) абзац по вашему вкусу в рамках границ текста на бумаге: отступы справа, слева, сверху, снизу; первая («красная») строка; положение текста в абзаце: по центру, прижать вправо, влево, по ширине
Формат, табуляция	Настраивает отступы для печати таблиц (табуляция), которые можно устанавливать в строке нажатием клавиши Tab
?	Осуществляет помощь, выводит подсказку

Несложно заметить общность всех инструментальных возможностей с учетом, что в разных редакторах они имеют различные по полноте реализации возможности.

Проектное задание

Познакомьтесь с клавиатурным тренажером, например в коллекции www.school-collection.edu.ru (раздел «Информатика», 7—9 классы, тренажер «Руки солиста»). Освойте расстановку пальцев рук для десятипальцевого метода печати.

4.2. Текстовый процессор

Текстовый процессор Word[1] обладает большими возможностями для формирования деловых бумаг, статей, научных публикаций и даже книг, готовых к изданию (рис. 4.3). Фактически это среда моделирования текстов различного вида по настройкам для каждого инструмента обработки текста со стороны пользователя. Управляя этими на-

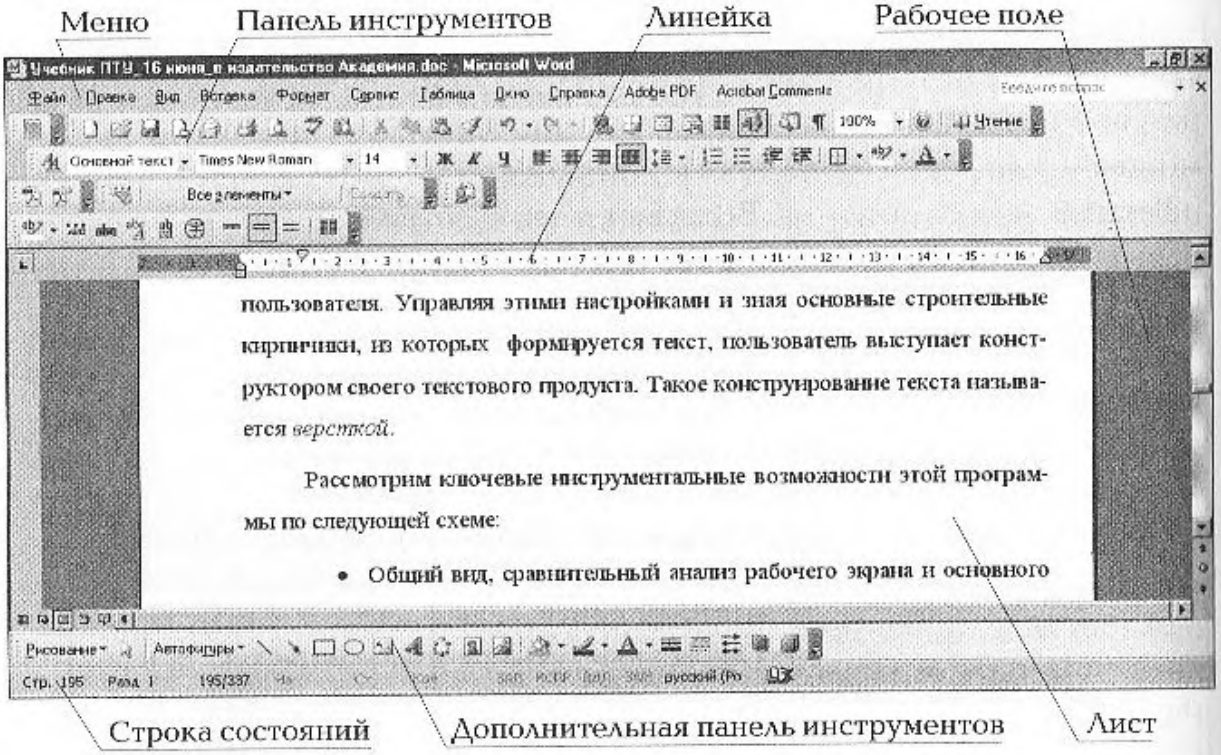


Рис. 4.3. Пример окна программы MS Word

[1] В зависимости отверсий ПО MS Office вид всех элементов окна может изменяться, а количество инструментов — увеличивается. *Базовые инструменты остаются без изменений.*

стройками и зная основные строительные кирпичики, из которых формируется текст, пользователь выступает конструктором своего текстового продукта. Такое конструирование текста называется *версткой*.

Основные инструментальные возможности этой программы включают в себя следующие элементы:

- общий вид, сравнительный анализ рабочего экрана и основного меню редакторов «Блокнот», WordPad и Word;
- общие команды работы с файлом в меню **Файл**, команда **Отменить** в меню **Правка** и меню **Помощь**, которые рассмотрены выше;
- параметры страницы;
- настройки абзаца;
- проверка правописания;
- фрагменты;
- шрифт, заголовки и буквицы;
- таблица;
- рисование;
- вставка объектов;
- оглавление.

Общий вид задается соответствующей услугой меню. Она присутствует во всех средах интегрированной среды и позволяет выкладывать на рабочий экран те или иные инструменты обработки текста с помощью команды **Панель инструментов**. Кроме того, в меню **Вид** можно настроить способ просмотра текста на рабочем экране редактора и размер компьютерного листа. Таким образом, уменьшив размер и установив способ просмотра «Разметка страницы», можно видеть на экране сразу несколько страниц своего документа издалека, наблюдая расположение текста на них.

Рабочий экран текстового редактора представляет собой *макет бумаги*. Характеристиками листа бумаги являются *параметры страницы*, т.е. размеры по вертикали и горизонтали и расположение, или ориентация. Эти настройки располагаются в меню **Файл**.

Стандартный размер бумаги для делопроизводства в нашей стране — А4, имеющий 210 мм по горизонтали и 297 мм по вертикали. Размер бумаги должен соответствовать параметрам устройства вывода. Если ваш принтер имеет габаритные размеры А4, то вы сможете печатать на нем документы данного размера и менее.

На предложенном на рис. 4.4 листе найдите верхнее, нижнее, правое и левое поля, поле переплета. Сравните левую и правую страницы в любой книге, сделайте вывод о зеркальных полях.

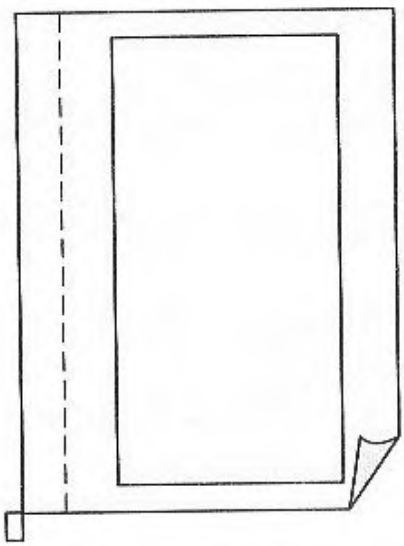


Рис. 4.4. Макет страницы

Абзац в текстовом редакторе — это часть текста, начинающаяся с «красной» строки и заканчивающаяся нажатием клавиши **Enter**. «Красная» строка может быть оформлена выступом, отступом или оставаться без оформления. Настройки абзаца заложены в меню **Формат**. Они содержат отступы слева и справа от границ текста на бумаге и отступ первой строки, которая является «красной». Можно использовать стандартное расстояние, заложенное в редакторе, а можно установить свое.

Интервалы между абзацами задаются высотой букв в строке, а межстрочные интервалы позволяют регулировать расстояние между строками в соответствии с высотой букв в строке: одинарный, полуторный и пр.

Пока не нажата клавиша **Enter**, вы находитесь в рамках данного абзаца, в котором будут выполняться все настройки. При переходе на следующий абзац можно установить новые настройки.

Внутри абзаца производится позиционирование текста: его можно прижать влево, вправо, центрировать и растянуть по ширине абзаца. Пиктограммы этих инструментов часто вынесены в строку меню программы.

Абзацем может быть даже одна буква. Заголовок — это абзац, который настраивается обычно более крупными буквами и часто располагается в центре строки.

Меню **Сервис** позволяет управлять переносами, выбирать языки, вызывать встроенный редактор **Орфография** — программу проверки правописания.

Фрагментом является любая часть текста, выделенная специальным образом. Выделить фрагмент можно мышью, продвигаясь по нужной части текста с нажатой левой клавишей. Фрагмент можно удалять, копировать в любом месте текста, перемещать в другое место текста.

Меню **Правка**, а именно команды **Копировать**, **Вырезать** и **Вставить** (обратите внимание на пиктограммы для этих команд), дает возможность использовать промежуточную (временную) память среды текстового процессора. Ее называют *буфером* («карманом»). На пиктограмме команды **Вставить** нарисован «карман» с листком текста. Все эти команды можно сопоставить с такими рабочими инструментами ручной верстки, как «ножницы», «копирка» и «клей».

Удаление фрагмента («ножницы») производится с помощью меню **Правка**, **Вырезать** или выделения и нажатия клавиши **Delete**.

Для *копирования фрагмента* (копирка) необходимо забрать его в промежуточную память — услуга меню **Правка**, **Копировать** (Ctrl + Insert); подвести курсор к нужной позиции в тексте и восстановить из промежуточной памяти («клей») — команда **Вставка** (Shift + Insert). Промежуточная память (буфер) хранит последний попавший в нее фрагмент на протяжении всей работы в редакторе.

Перемещение фрагмента можно выполнить мышью без команд меню **Правка** с помощью технологии «хватай-тяни»: выделить фрагмент, установить указатель мыши на него и нажать левую клавишу; не отпуская клавишу, дождаться появления картинки прямоугольника, переместить мышь с нажатой клавишей в нужное место текста.

Шрифтом представлен способ изображения букв в печатном тексте. Шрифт обладает следующими характеристиками: вид, размер и стиль (начертание).

Шрифты делятся на несколько категорий:

- *экранные* или *немасштабируемые* — используются для вывода информации на экран, но при распечатке документа на принтере не гарантируется полная аналогия с экранным образом;
- *принтерные* — предназначены для печати на конкретном принтере. Программа не всегда правильно подбирает экранное отображение, поэтому напечатанный документ может быть искажен;
- *TrueType*, или *масштабируемые*, — для них гарантируется полное совпадение при печати.

В списке шрифтов перед названием принтерного шрифта стоит значок принтера, а перед TrueType-шрифтом — двойная буква Т. При выборе шрифтов не следует назначать более трех разновидностей в одном документе.

Вот некоторые популярные шрифты:

- Arial Cyr — стандартный шрифт Excel;
- Times New Roman — стандартный шрифт Word (приближен к типографскому шрифту);
- Bodoni — приближен к типографскому шрифту;
- Courier — используется для печати корреспонденций;
- Symbol — греческие символы;
- Wingdings — разнообразные значки вместо букв.

Кроме названия можно указать *начертание* (стиль) шрифта: нормальное, курсивное, полужирное (жирное) и полужирный курсив. Размер шрифта измеряется в *пунктах* (1 пункт = 1/72 дюйма = 0,357 мм). Вид шрифта определяется его названием. Размер задается из перечня

размеров или вручную, а начертание стандартно предлагается любому виду шрифта: обычное, жирное и наклонное (курсив).

Проведите опыт с этими характеристиками с помощью команды **Шрифт** в меню **Формат**. В окне образца вы увидите, как изменяется вид букв при выборе различных характеристик шрифтов.

Особое внимание при просмотре видов шрифтов обратите на то, имеют ли они русские буквы. Не все шрифты русифицированы. Кроме того, есть шрифты, содержащие не буквы, а символы и даже картинки. Познакомьтесь с ними. С их помощью можно украсить вашу визитку. Установить в текст отдельный символ позволяет команда **Символ** в меню **Вставка** (рис. 4.5).

Кроме этих основных характеристик есть дополнительные возможности управления оформлением букв в тексте: подчеркивание (всевозможные его виды можно посмотреть в настройке **Подчеркивание** в команде **Шрифт**), эффекты, в которых особое внимание следует уделить верхнему и нижнему индексам, они позволят устанавливать в тексте такие объекты, как x_i и x^n (табл. 4.4).

При оформлении в тексте особо важных *замечаний* их выделяют разрядкой, т.е. печатают слово с широким расстоянием между буквами. Это расстояние можно настроить в настройке **Интервал**.

Представляет интерес украшение текста буквицей. *Буквицей* называется первая буква абзаца, оформленная специальным шрифтом, размером и стилем, для яркого и броского выделения в тексте. В древних книгах буквицу украшали растительным орнаментом, она представляла собой произведение искусства, живописную миниатюру. Выделив нужную букву в тексте подсветкой, следует обратиться к команде **Буквица** в меню **Формат** и выбирать необходимые настройки.

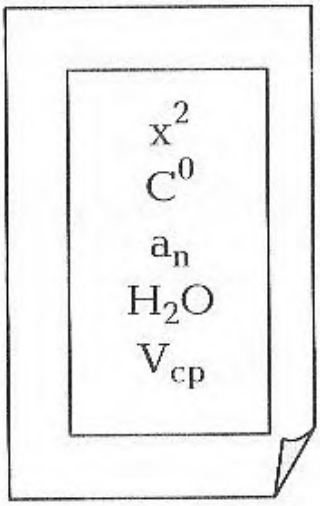


Рис. 4.5. Пример вставки символов в выражения

Таблицы являются неотъемлемой частью текста. Благодаря им можно представить большой объем информации емко и наглядно. Таблица — это форма представления информации по столбцам и строкам. Каждый столбец содержит информацию одного смысла. Заголовки столбцов образуют «шапку» таблицы. Нужно различать понятие таблицы и текста, ее заполняющего. Таблица — способ организации информации. Она создается как пустая конструкция. Ее наполнение информацией — это процесс набора текста в ячейках та-

Таблица 4.4

Команда меню «Шрифт»	Описание
Шрифт	Выбирает вид шрифта из библиотеки шрифтов на компьютере, ее можно пополнять с компакт-диска
Размер	Устанавливает размеры букв в шрифте
Начертание	Определяет стиль написания букв шрифта: жирный, курсив, подчеркивание
Подчеркивание	Дает образцы линий для подчеркивания слов
Видоизменение	Приводит способы оформления букв
Цвет текста	Устанавливает цвет букв
Интервал	Определяет расстояние между буквами в слове
Анимация	Украшает текст цветными движущимися акцентами, вызывающими особое внимание в тексте для компьютерного, но не бумажного использования

блицы. Каждая ячейка таблицы представляет собой свое поле набора текста, где работают все правила абзаца.

Для удобства работы с таблицами на экран можно выложить инструменты **Таблица и границы** из меню **Вид** командой **Панель инструментов**. Задается таблица с помощью команд меню **Таблица** (табл. 4.5).

Таблица 4.5

Команда меню «Таблица»	Описание
Нарисовать таблицу	Создает пустую таблицу с выбором количества строк и столбцов в ней
Выровнять (высоту и ширину)	Автоматически выравнивает размеры строк и столбцов, формирует размеры клеток, строк и столбцов
Выделить	Выделяет подсветкой фрагмент таблицы (ячейки, столбцы, строки)
Удалить	Удаляет часть таблицы
Вставить	Вставляет строки, столбцы в ту часть таблицы, куда установлен курсор
Объединить, разбить (ячейку)	Осуществляет слияние нескольких ячеек таблицы в одну или разделение одной ячейки на несколько

Окончание табл. 4.5

Команда меню «Таблица»	Описание
Сортировка	Упорядочивает по возрастанию или убыванию информацию в таблице
Разбить (таблицу)	Делит одну таблицу на две по месту нахождения курсора
Отобразить или скрыть сетку	Прорисовывает разделительные линии строк и столбцов, которые называются сеткой таблицы
Автоформат таблицы	Украшает таблицу обрамлением, сеткой и цветовыми пометками. Предлагаются всевозможные стили автоматического украшения таблицы, выбрав один из которых можно быстро получить оформленную таблицу в соответствии с предложенным образцом

Раскрасить цветом нужные ячейки можно, выделив их подсветкой и воспользовавшись командой **Границы и заливка** в меню **Формат**.

Иллюстрации в тексте (картинки, рисунки, схемы) можно устанавливать в тексте с помощью меню **Вставка** и инструментов встроенного рисования в редакторе.

Сначала познакомимся с возможностями **рисования**, предложенными самой средой редактирования. Для этого необходимо открыть **Портфель инструментов** в меню **Вид**, **Панели инструментов**, а затем «выложить» на экран для удобной работы инструменты **Рисование** и **Настройка изображения**.

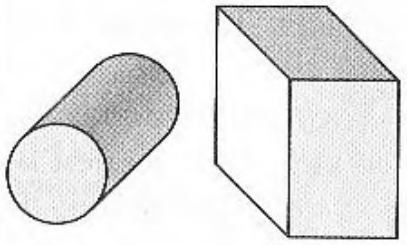
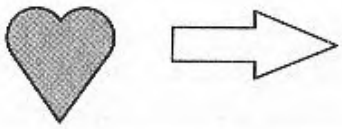
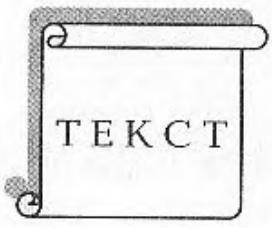
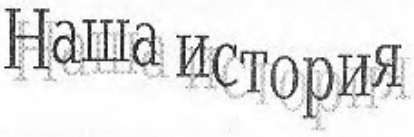
Все рисованные объекты имеют рамку деформации, потянув за узлы которой, можно изменить размер изображения и деформировать его, сжимая и растягивая. При этом если рамка содержит желтый узелок, то можно изменять форму изображения.

Особо отметим положение картинки в тексте (табл. 4.6).

> В **Настройке изображения** содержится команда **Обтекание**. Познакомьтесь с видами обтекания. Рассмотрите иллюстрированные страницы любого учебника и постарайтесь определить, каков вид обтекания для них был выбран. Параметр **Без обтекания**, выбранный для рисунка, позволит сделать рисунок фоном, на котором можно установить текст.

В текстовых редакторах можно осуществлять **вставку объектов** (иллюстраций) из имеющихся коллекций. Эти коллекции или поставляются вместе с программной средой, или размещаются в других файлах, в том числе на дисках, в Интернете. Рассмотрим предложенные

Таблица 4.6

Команда меню «Рисование»	Описание	Дополнительные возможности
Квадрат, прямоугольник, круг, овал	Рисует на листе геометрические фигуры круг и прямоугольник, при этом их можно деформировать	Рисует тенью, устанавливает объем 
Отрезок и стрелка	Рисует прямую линию или линию со стрелкой на конце	Определяет толщину линии, вид линии, вид стрелки 
Автофигуры: линии основные фигуры блок-схемы звезды и ленты фигурные стрелки	Выбирает прямые и кривые и пр.	Раскрашивает, настраивает по отношению к тексту, вращает 
Действия (порядок)	Устанавливает на передний (задний) план, перед текстом, за текстом	Оформляет свиток с тенью за текстом 
Коллекция WordArt	Выбарает плакатные стили для заголовков	Вращает, устанавливает форму надписи, цвет и пр. 
Вставка, надпись	Устанавливает текст в любом рисунке	Вращает, определяет шрифт, цвет

в меню коллекции WordArt, диаграммы, автофигуры. Объекты этих коллекций оказывают помощь в подготовке плакатов и схем, которые также являются текстовыми документами. Ниже описаны инструменты работы с ними.

Рисунок попадет из коллекции или выбранного файла в текст, если выбрать с помощью мыши и указать команду **Вставить**. Однако рисунок нужно внедрить в текст так, чтобы он не нарушил форматирование в абзацах. Для этого используют настройки рисунка: *размер*, *обтекание*, *цвет*.

Рисунок расположится в окружении рамки с узелками настройки, если щелкнуть на нем мышью. Узелками регулируется размер, производится деформация. Для *вращения* и *поворота* нужно использовать те узелки, которые поменяют цвет, а обтекание настроит рисунок на установку его в тексте в выбранном месте. Далее потребуется только ваше желание использовать эти инструменты.

? Вопросы и задания

1. Какие функции текстового редактора и процессора можно определить как общие? Воспользуйтесь для ответа редакторами «Блокнот» и Word.
2. Какие инструментальные средства меню предусмотрены для моделирования текста?
3. Что такое абзац как строительный материал текста? Какие виды настроек предусмотрены для абзаца?
4. Какие ключевые инструменты для моделирования текста имеются в редакторе Word? Каковы возможности этих инструментов и как они помогают моделировать текст?
5. Что такое интегрированная программная среда? Обоснуйте свой ответ на основе стандартных приложений Microsoft Windows или Open Office.

П-Г Проектное задание

Постройте модели «Афиша» и «Театральная программка». Подготовьте план оформления афиши и программки в текстовом редакторе. Для программки используйте инструменты «Таблица» или двухколонный текст, причем ориентацию бумаги возьмите альбомную. Тогда можно будет сложить программку пополам, а на внешней стороне распечатать рисунок к ней. Вставку рисунка можно сделать из файла, приготовленного самостоятельно, например в редакторе Paint, или из Интернета.

Для афиши используйте портрет выступающего, который можно отсканировать с фотографии.

Распечатайте программку на принтере и далее можете использовать копировальный аппарат для получения нужного количества экземпляров. Афишу рекомендуется выполнить в электронном виде и разместить на сайте образовательного учреждения в разделе мероприятий.

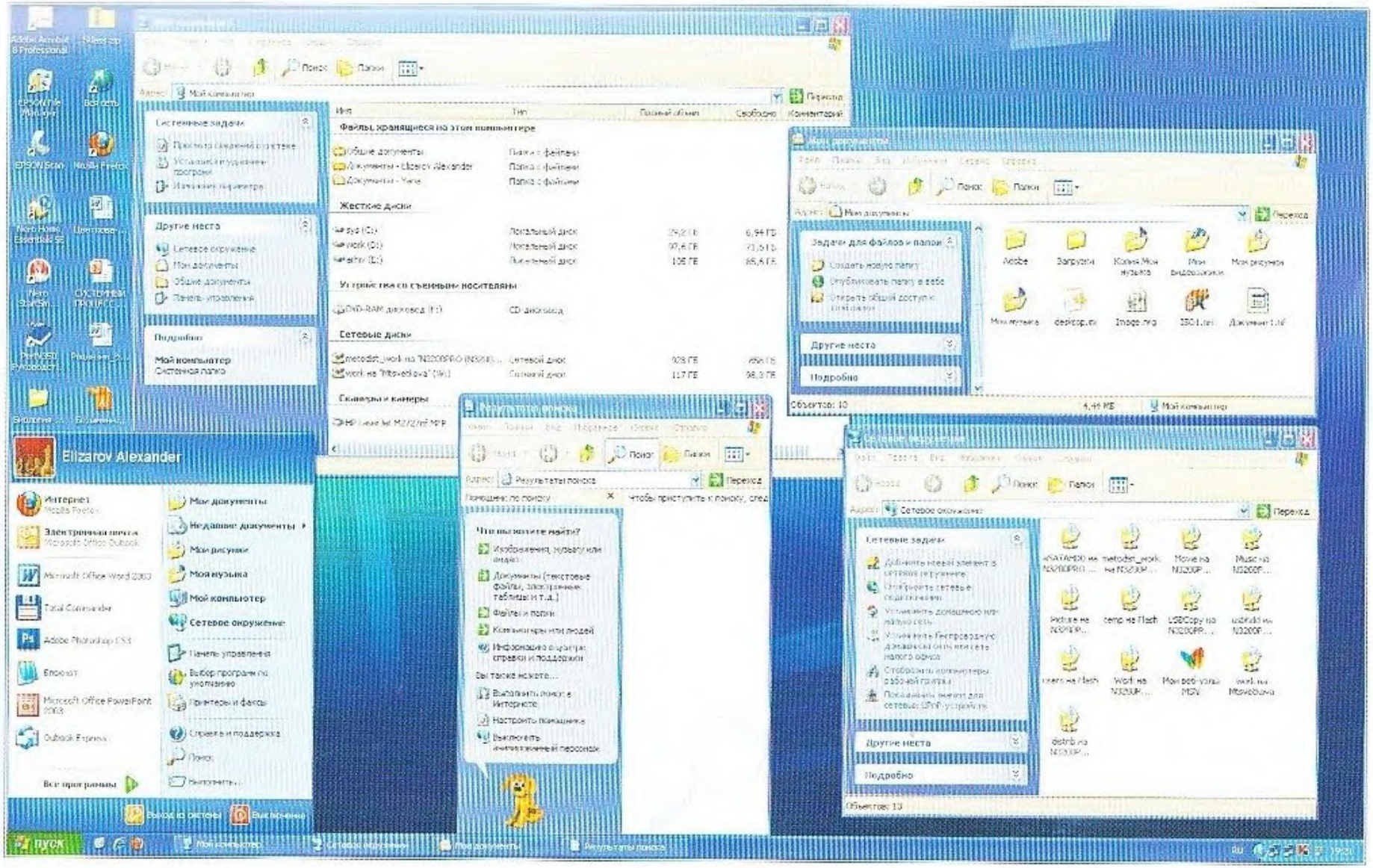


Рис. 1. Вид «Рабочего стола» в ОС Microsoft Windows

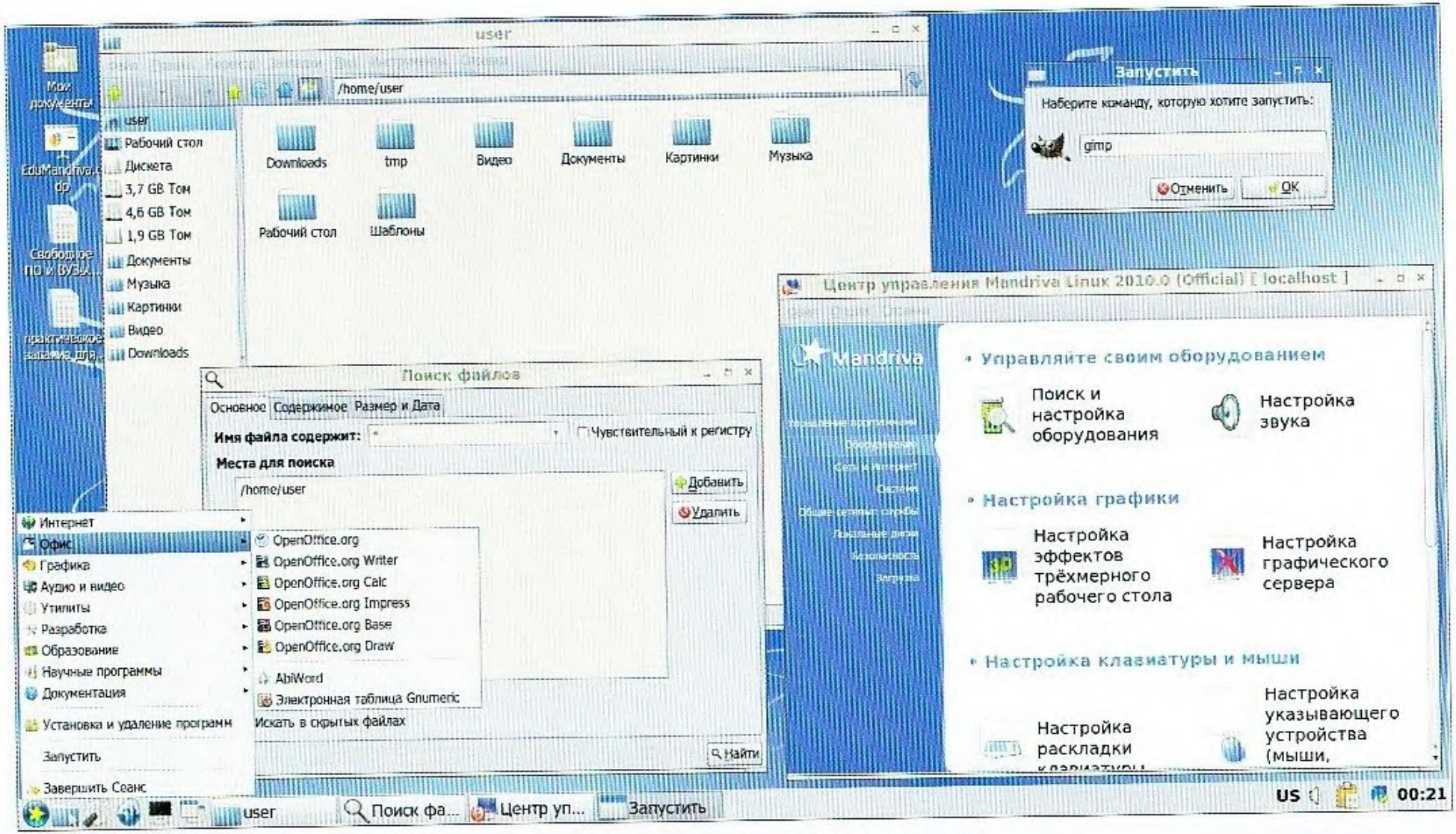


Рис. 2. Вид «Рабочего стола» в ОС Mandriva Linux

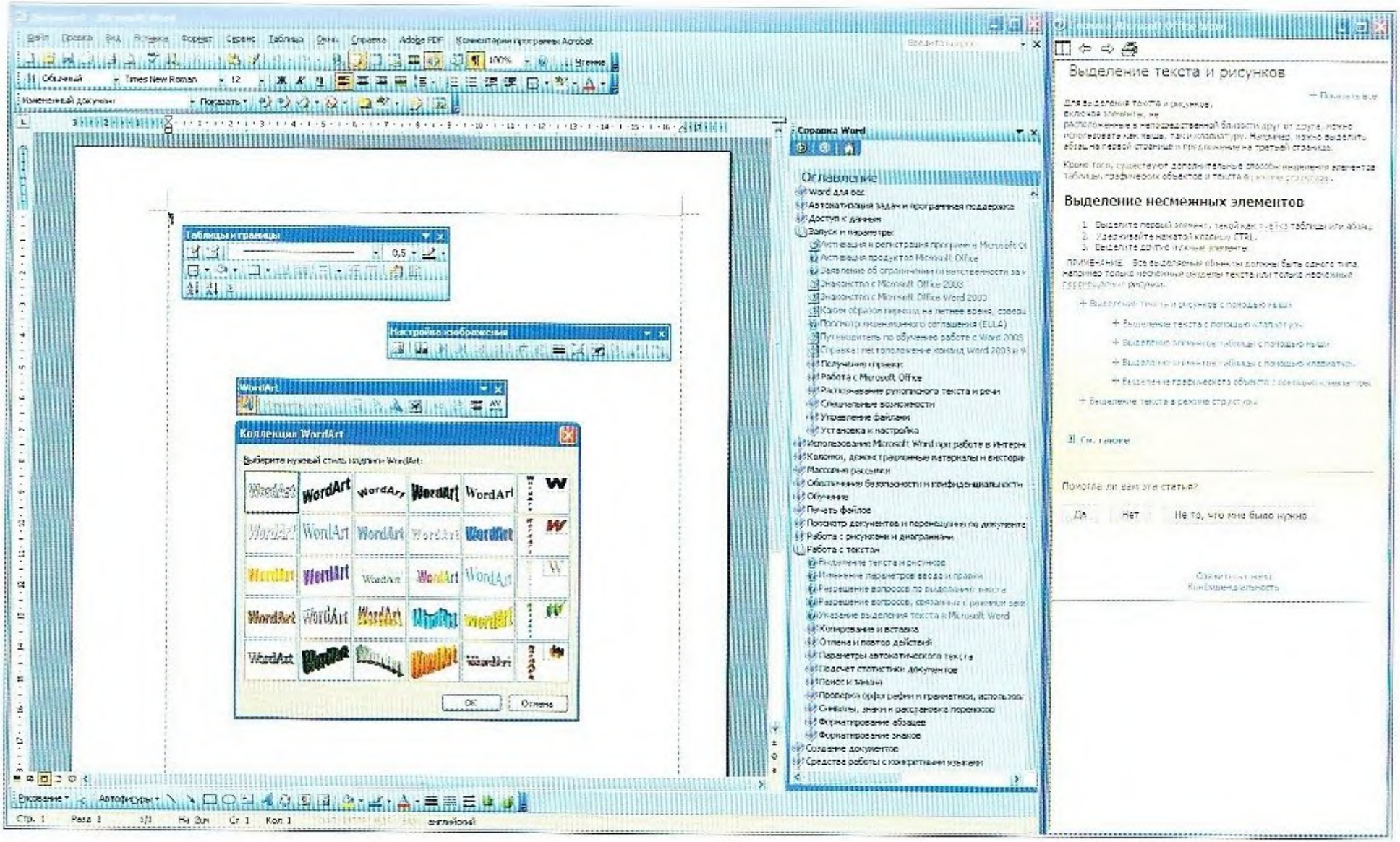


Рис. 3. Вид окна текстового редактора Microsoft Word с открытым режимом «Справка»

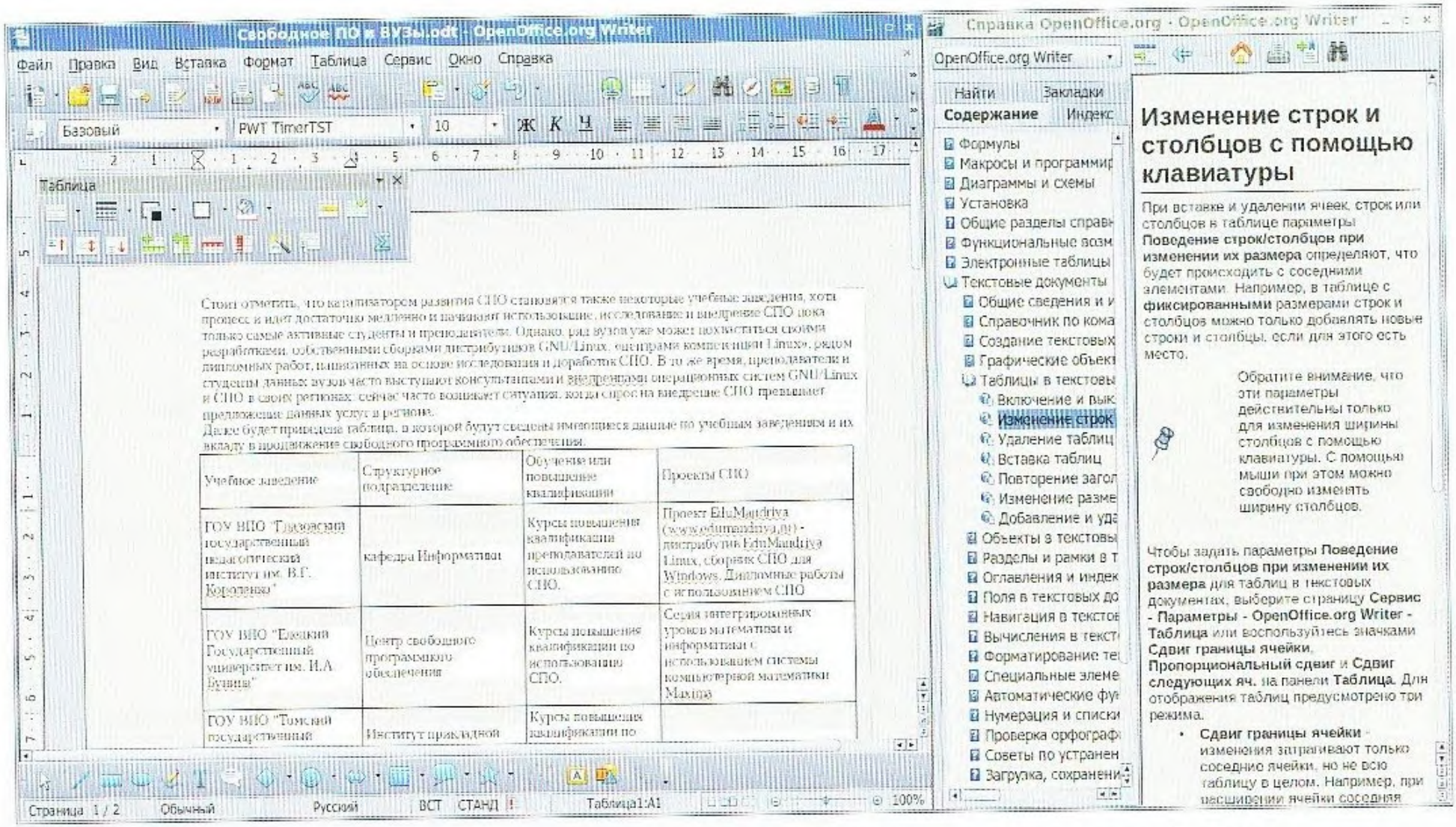


Рис. 4. Вид окна текстового редактора OpenOffice.org Writer с открытым режимом «Справка»

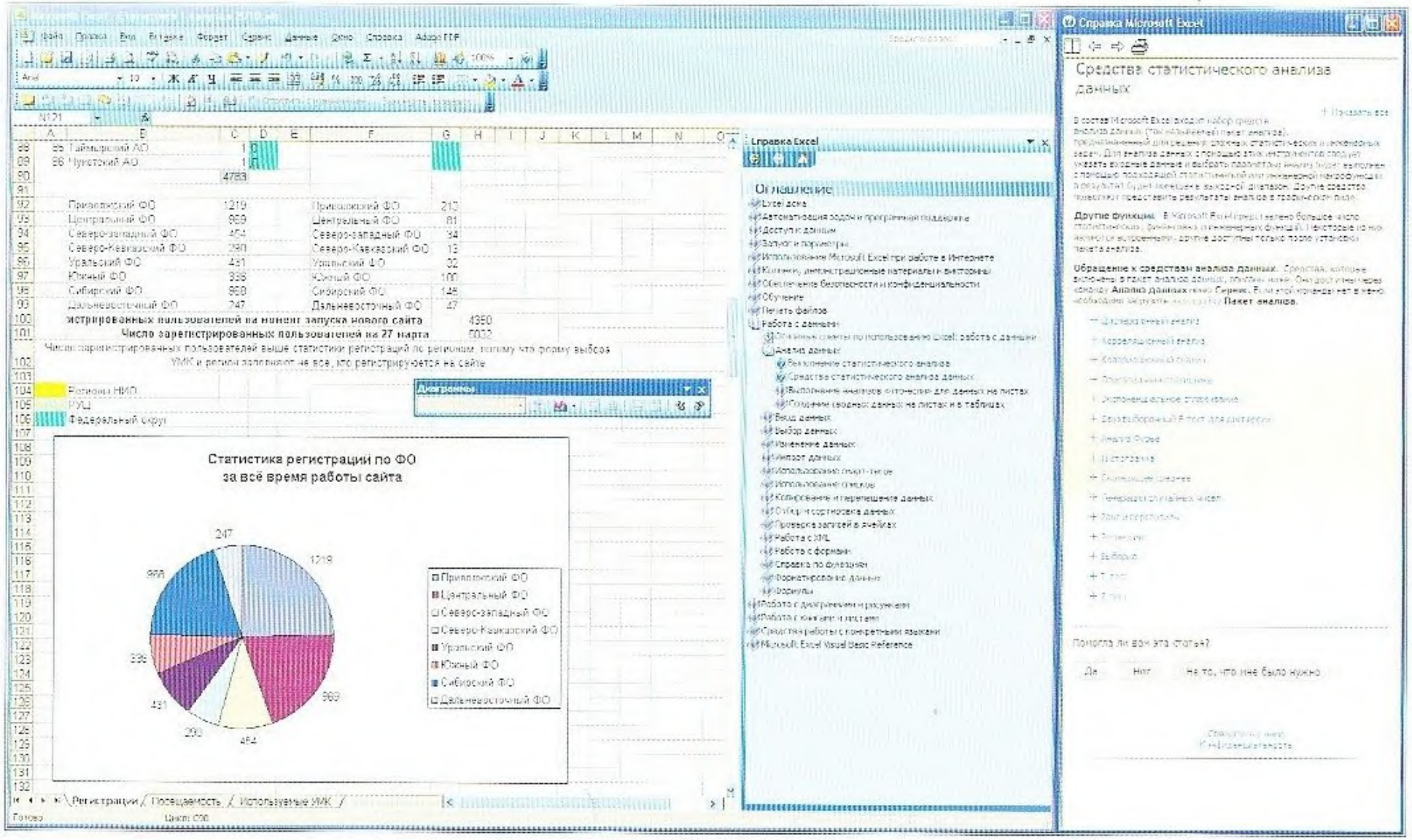


Рис. 5. Вид окна электронной таблицы Microsoft Excel с открытым режимом «Справка»

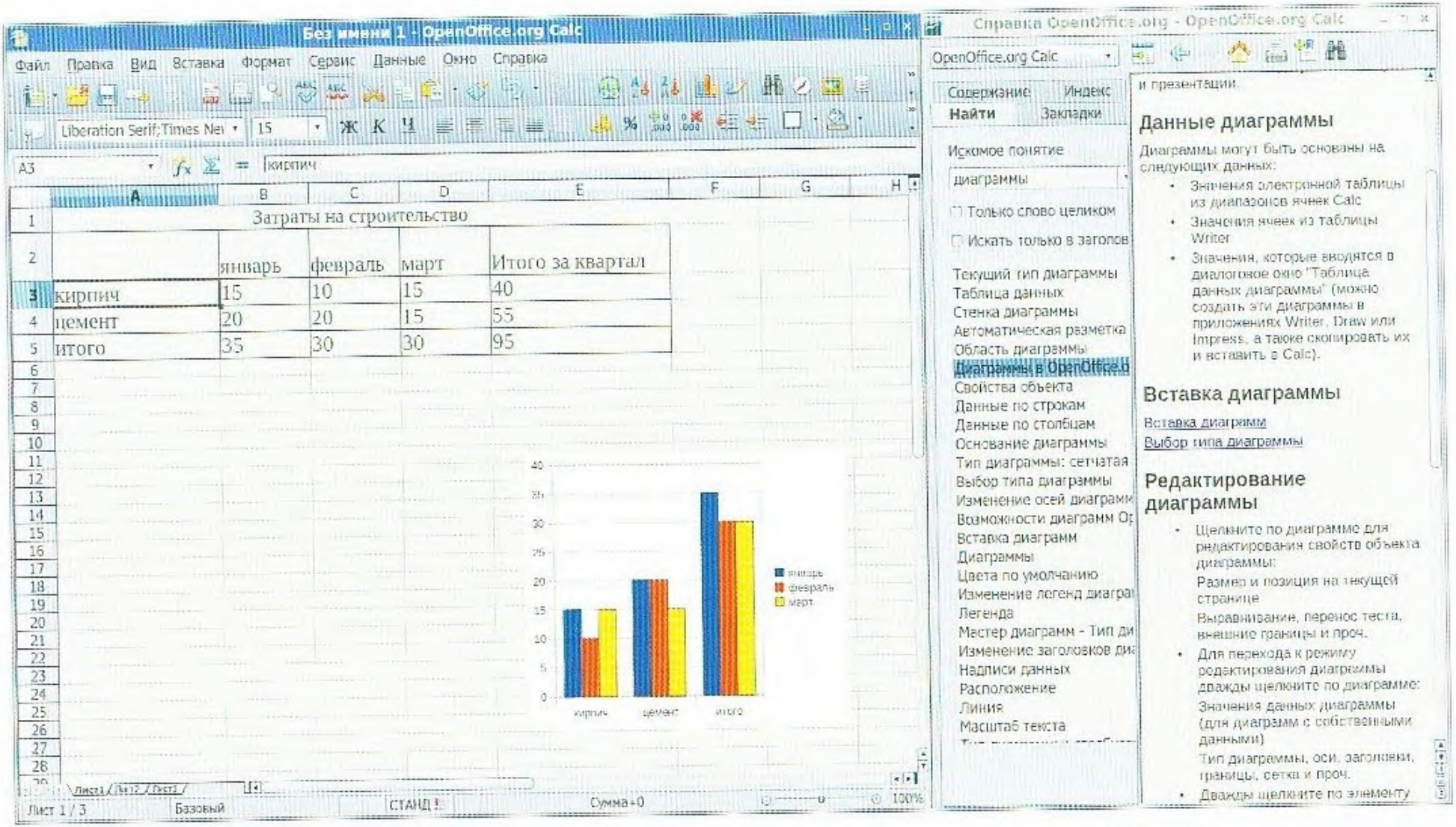


Рис. 6. Вид окна электронной таблицы OpenOffice.org Calc с открытым режимом «Справка»

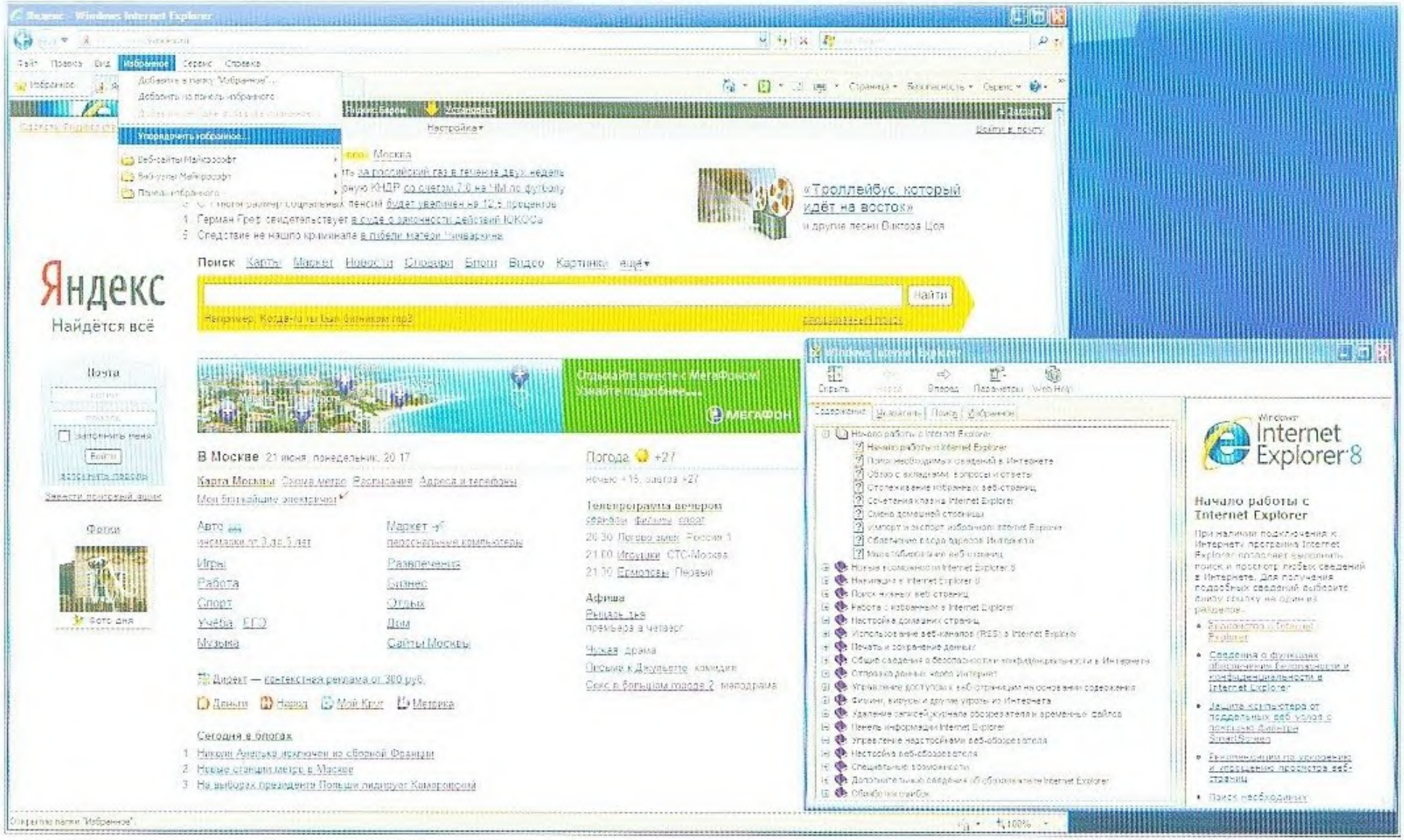


Рис. 7. Вид окна интернет-браузера Microsoft Internet Explorer с открытым режимом «Справка»

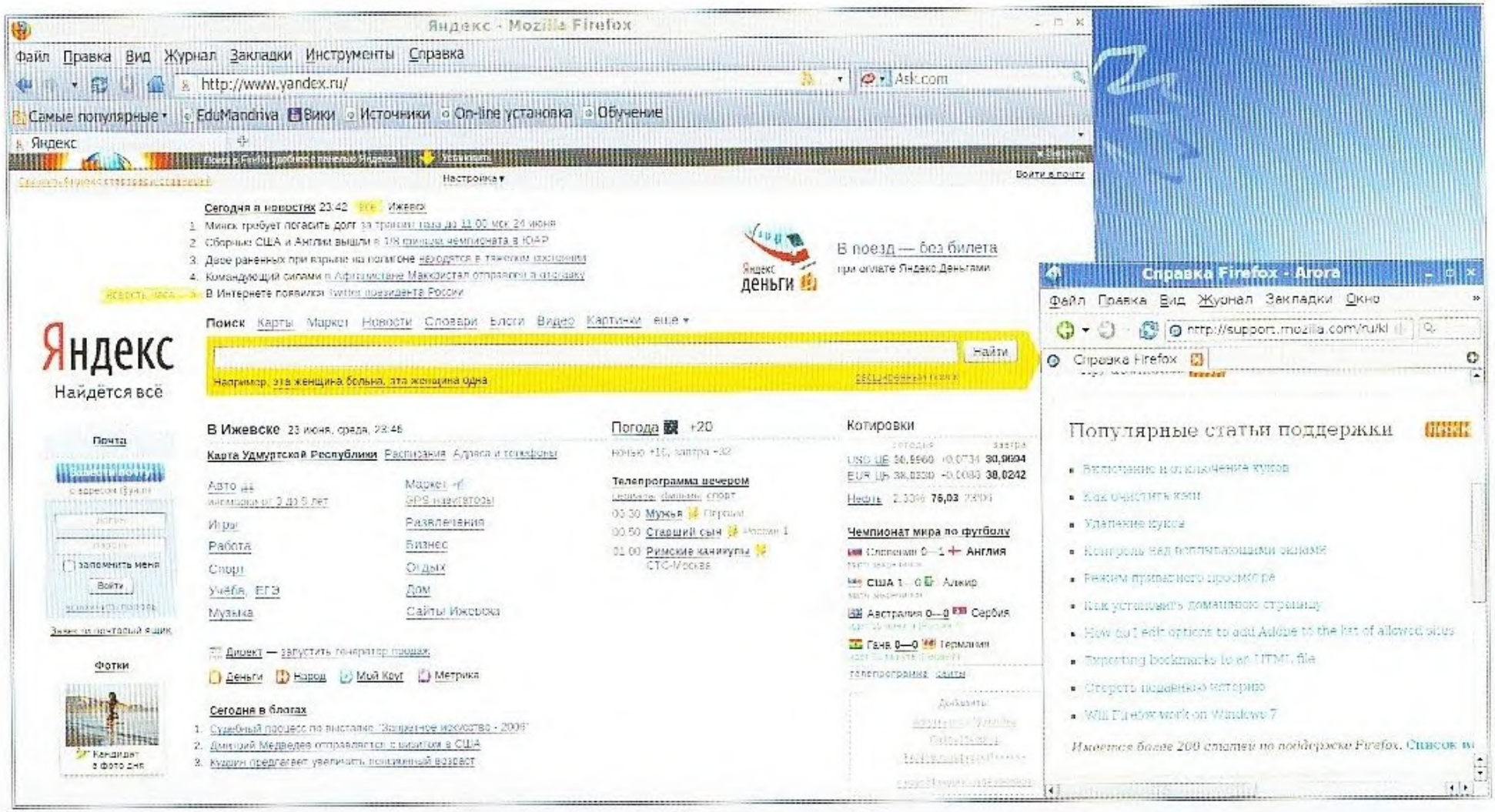


Рис. 8. Вид окна интернет-браузера Mozilla Firefox с открытым режимом «Справка»

4.3. Использование шаблонов документов и других средств, повышающих эффективность работы с текстом

Вы познакомились с основными инструментами формирования документа. Рассмотрим теперь дополнительные команды управления текстом.

Шаблон — это особый вид документа, представляющий собой модель издания и специальные средства для оформления итогового документа по типовой схеме. Шаблон может содержать следующие элементы:

- текст или формат, одинаковый для всех документов этого типа, например, для служебной записки или отчета;
- стили;
- элементы автотекста;
- макросы;
- меню и присвоенные сочетаниям клавиш операции;
- панели инструментов.

Шаблон «Обычный» является универсальным шаблоном, подходящим для разных типов документов. При запуске программы Word или нажатии кнопки **Создать** создается новый пустой документ, основанный на шаблоне «Обычный». Этот шаблон можно редактировать, чтобы изменить стандартное форматирование или содержимое документа. Существуют шаблоны для создания служебных записок, писем, различного рода резюме, информационных бюллетеней, факсов, отчетов, публикаций и т. д. Шаблон во многом определяет стиль документа и зачастую его успешность. Например, правильно составленное резюме безусловно повышает шансы приема на работу его автора при прочих равных условиях. Используя шаблон резюме в Word, его достаточно заполнить, поскольку вы получаете в работу готовую модель этого документа.

Система шаблонов по различным видам документов также предлагается в специальной программной среде MS Publisher, в которой представлены шаблоны газет, дипломов, буклетов и даже страницы сайта для интернет-газеты.

Электронный текст с **гиперссылками** в отличие от текста, предназначенного для бумажного тиражирования, кроме всевозможных оформительских эффектов позволяет оформить быстрые переходы к нужной информации. Если вам при работе с бумажной книгой придется искать нужную страницу по оглавлению, то компьютерная книга перешлет вас к этой странице автоматически. Для этого в тексте

Таблица 4.7

Команда меню «Вставка»	Описание
Разрыв	Устанавливает в документе новую страницу
Номера страниц	Расставляет страницы на листах в указанном месте, начиная с указанного номера страницы
Дата и время	Устанавливает дату и время по образцу
Перекрестная ссылка	Устанавливает ссылку с автоматическим переходом в нужное место текста: на страницу, абзац
Оглавление и указатели	Устанавливает в тексте оглавление с автоматической сборкой указанных заголовков по уровням подчиненности, осуществляет сборку оглавления рисунков, таблиц
Гиперссылка	Устанавливает связь документа с другими файлами

используются гиперссылки или инструмент «Оглавление». Гиперссылки можно установить в тексте с помощью команды меню **Вставка**, **Перекрестная ссылка**.

Рассмотрим создание оглавления в автоматическом режиме (табл. 4.7). Это полезно научиться делать для оформления рефератов.

Наберите текст реферата. Расставьте номера страниц с помощью меню **Вставка** командой **Номера страниц**.

Выделенный абзац, содержащий текст заголовка, необходимо определить стилем *уровня заголовка*. Номер уровня показывает степень его важности.

Когда все заголовки назначены, можно приступать к сборке оглавления (табл. 4.8). Для этого в меню **Вставка** необходимо выбрать команду **Оглавление и указатели**.

Таблица 4.8

Настройки команды «Оглавление и указатели»	Описание
Оглавление	Переход в режим создания оглавления
Вид	Выбор шаблона оглавления из показанных в образце

Окончание табл. 4.8

Настройки команды «Оглавление и указатели»	Описание
Уровни	Указание, до какого уровня заголовков формируется оглавление, т.е. можно включать в оглавление не все уровни заголовков
Иллюстрации	Переход в режим создания оглавления рисунков и таблиц

Пусть ваш реферат содержит две главы. Названию каждой главы назначьте уровень 1 заголовка. Глава содержит насколько разделов. Их названиям назначьте уровень 2 заголовка. В параграфе может быть пункт. Его названию назначьте уровень 3 заголовка. Установите выделение в заголовках реферата и назначьте им уровень заголовка с помощью команды **Стиль** в меню **Формат**. Можно сделать это быстрее, используя меню стилей в верхней части экрана (рис. 4.6).

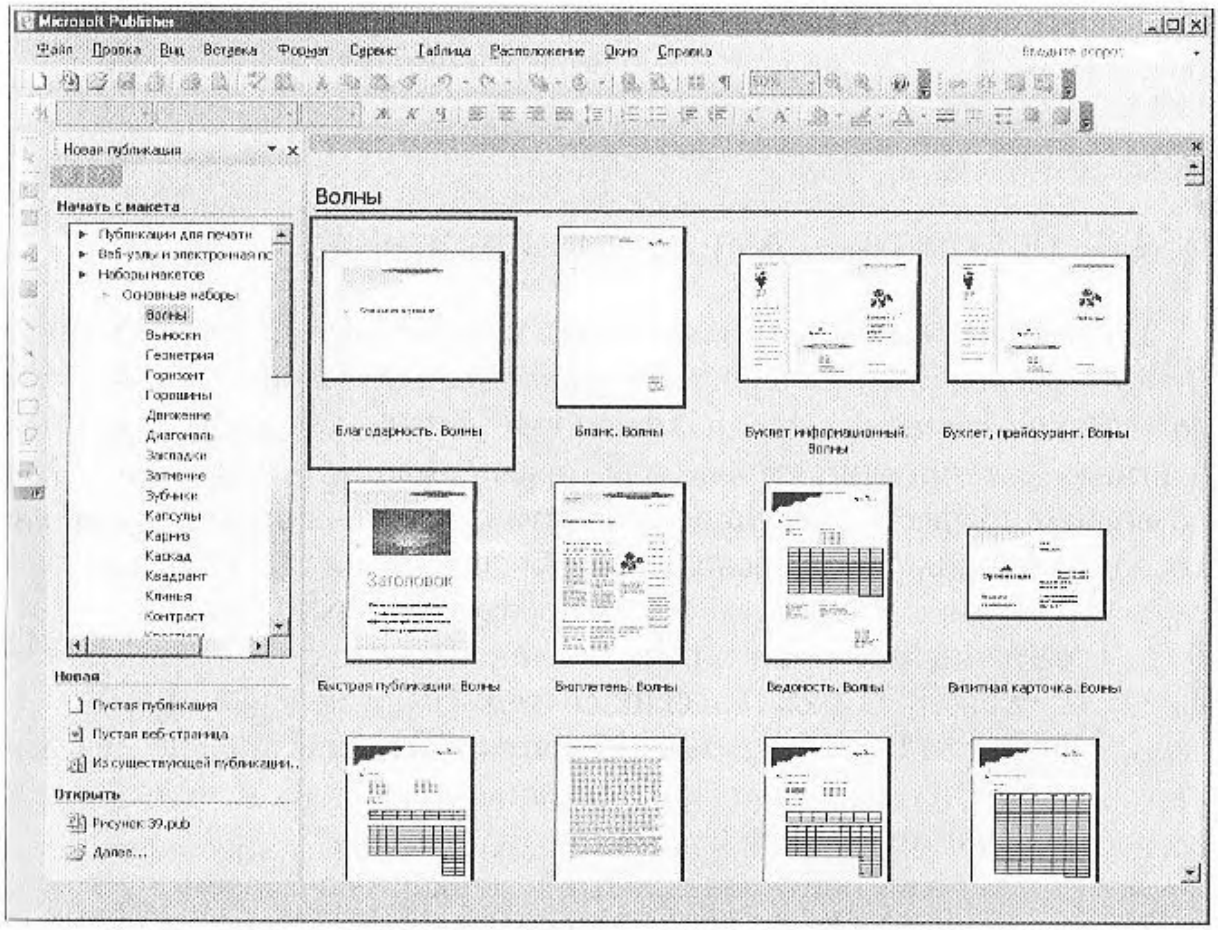


Рис. 4.6. Окно макетов (шаблонов) программы MS Publisher

Оглавление можно переделать в любой момент. Для этого следует установить курсор на оглавлении и вновь выполнить команду **Оглавление и указатели**, выбрав затем действие **Заменить**. Аналогично можно сделать оглавление рисунков, таблиц.

Вставьте ссылки на литературу. Наберите список литературы (рисунков или таблиц), а затем в тексте установите ссылки, используя команду **Перекрестные ссылки** и настройки по типам ссылок (абзац, рисунок, таблица) с выбором необходимой книги (рисунка или таблицы) из списка с помощью мыши.

? Вопросы и задания

1. Что представляет собой шаблон? В каких случаях он используется?
2. Что такое гиперссылка? В каких случаях она используется?
3. Какие команды содержит меню **Вставка**? Охарактеризуйте каждый элемент.
4. Используя шаблоны Word, оформите свое резюме, визитную карточку.

П Проектное задание

Выполните проекты: «Реферат» и «Статья для конференции». Оформите расстановку страниц, сборку оглавления, список литературы, установите сноски в тексте. Вставьте рисунки, таблицы, оформите схемы с помощью автофигур.

4.4. Программы для верстки оригинал-макетов

Текстовые процессоры применяются в полиграфии и книгоиздании. Для этих целей имеются профессиональные программные среды — *издательские системы*, предназначенные для верстки электронных оригинал-макетов книг, готовых для тиражирования в типографии. Они позволяют профессионально подготовить к публикации книгу, учебное пособие, сборник стихов, рекламный проспект и другую полиграфическую продукцию. Попробуем себя в профессии верстальщика.

Для создания различного рода публикаций в настоящее время существует целый ряд издательских систем, среди которых можно выделить Publisher, PageMaker, QuarkExpress, Corel Ventura и др. Все эти программные продукты русифицированы, так что их использование не представляет большой сложности.

Для разработки оригинал-макета публикации выберем издательскую систему Adobe PageMaker. Демонстрационную версию этой системы можно получить на официальном сайте компании *Adobe*: www.

adobe.com. Для этого требуется с помощью браузера зайти на указанный сайт в раздел Downloads/Trial Downloads и выбрать интересующий вас продукт Adobe PageMaker. Далее необходимо зарегистрироваться на странице загрузки, чтобы получить возможность загрузить демоверсию программы для ее свободного использования в течение 30 дней.

Самое главное при создании публикации — определить параметры для модели вашего документа, такие как формат, ориентация страниц, размер полей и т. д. Для этого необходимо запустить программу PageMaker и выбрать в меню **Файл** (File) команду **Новый** (New) (рис. 4.7).

Далее следует выполнить подготовку оригинал-макета, задав необходимые параметры страниц в появившемся окне «Параметры документа» (Document Setup). Для этого требуется выполнить следующие действия:

1) в области «Формат» (Page Size) выбрать один из предлагаемых стандартных вариантов бумаги, например A4. При этом размеры страниц будут автоматически показаны в полях окна экрана в области «Размер» (Dimensions);

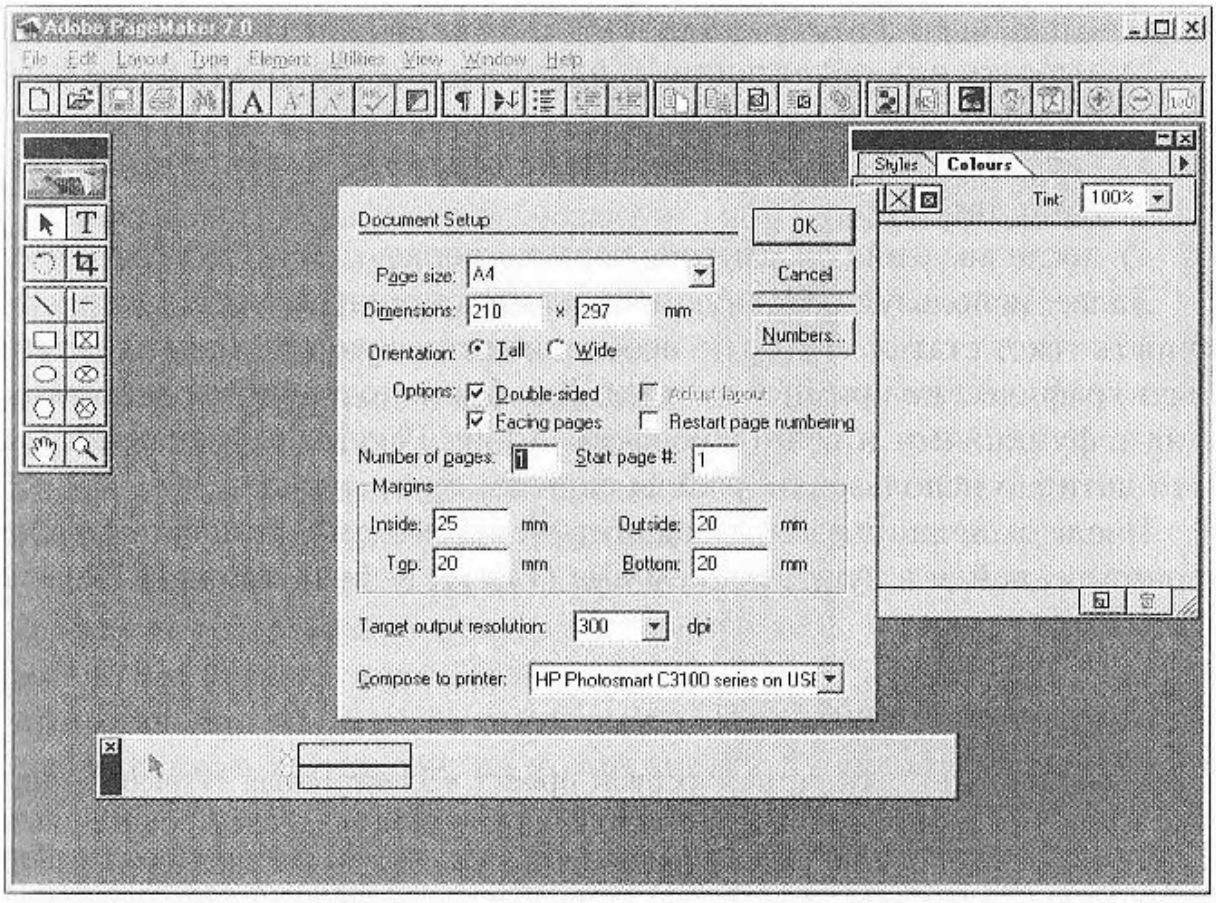


Рис. 4.7. Задание параметров страницы в Adobe PageMaker

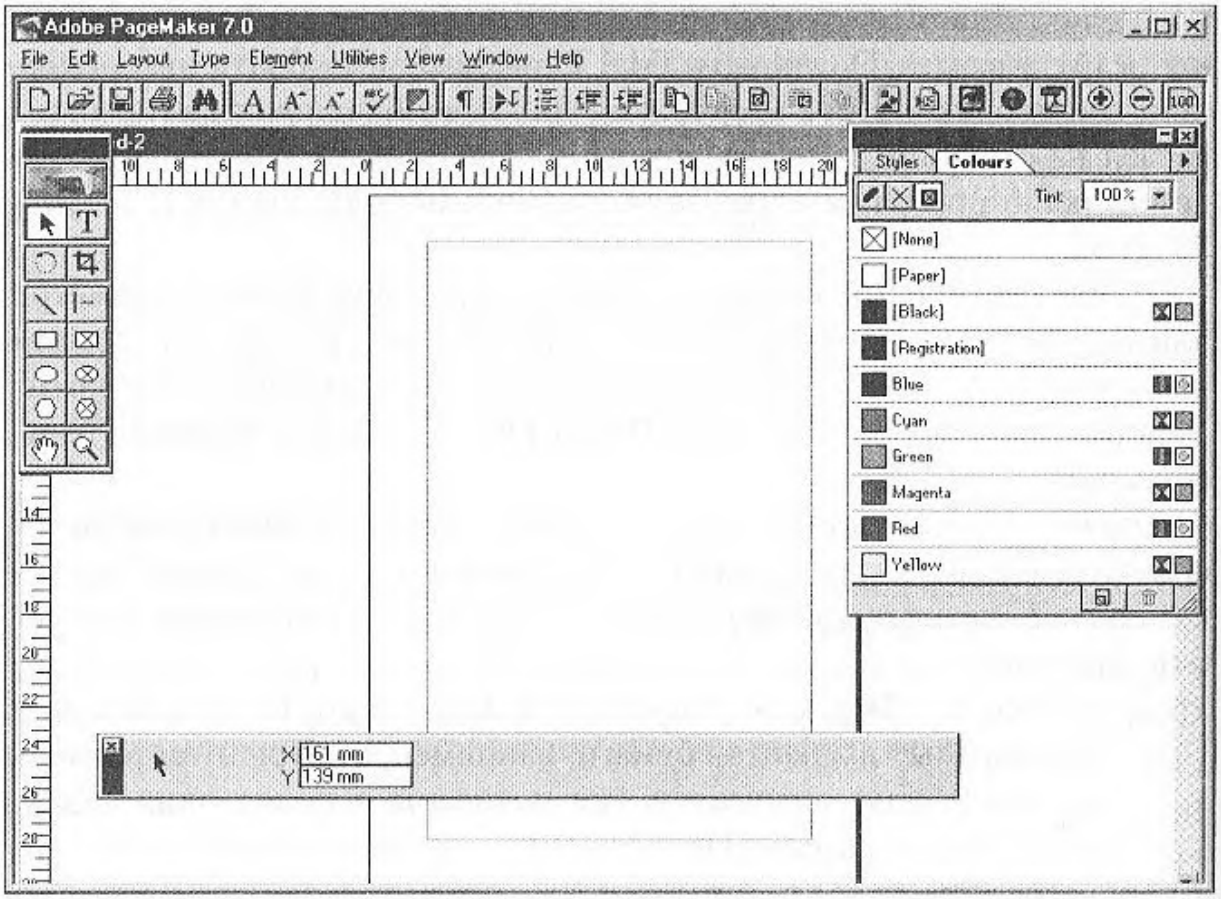


Рис. 4.8. Создание публикации в Adobe PageMaker

2) выбрать вид ориентации «Книжная» (Portrait/Tall) или «Альбомная» (Landscape/Wide);
3) ввести размеры полей текста на странице в области «Поля» (Margins);
4) после введения параметров нажать кнопку «ДА» (OK) (рис. 4.8).

Далее, используя основную панель инструментов, вы сможете набирать текст статьи, меняя по необходимости шрифты и цвета, вставлять графические примитивы и т. д. Для более подробного ознакомления с функциями и возможностями Adobe PageMaker рекомендуем вам активно использовать файлы справки программы.

После окончания работы над статьей вы сможете распечатать ее на принтере, войдя в пункт меню **Файл** (File) и выбрав команду **Печать** (Print).

Курсовой проект

Выполните коллективный курсовой проект «Электронная тетрадь по информатике», в котором отразите все задания из учебника в электронном виде по мере их накопления, разместив их на диске общего пользования. Такой курсовой проект позволит сформировать электронное приложение к учебнику в виде примеров решения всех заданий.

4.5. Технология обработки графической информации

Графические редакторы — это программы для работы с рисунком в графическом режиме работы монитора. С графическим экраном удобно работать с помощью мыши, компьютерного пера, графпланшета. Существует большое количество мощных программных средств для обработки изображений: Adobe Photoshop, Adobe Illustrator, CorelDRAW, Macromedia FreeHand, Inkscape.

Выясним общие принципы работы с изображением на компьютере с помощью простейшего графического редактора Paint, помня о том, что этот редактор из стандартного приложения Windows является частью интегрированной среды редакторов (рис. 4.9).

При входе в редактор и сравнении его основного меню команд с меню текстового редактора становится ясным, что команды меню **Файл**, **Вид** и **Помощь** остались неизменными. Как и во всех редакторских средах, нам необходимо работать с файлом, раскрывать или прятать на экране инструменты программы и получать помощь. Файл, который создается в графическом редакторе, имеет другой тип — это графический файл. Он содержит коды координат пикселов картинки и коды цветов этих пикселов. Сохранив свою первую картинку, нарисованную с помощью мыши, вы убедитесь, что создан файл, содержащий рисунок. Типы файлов мы обсуждали ранее, они могут быть такими: PCX, BMP, TIF и др.

Меню многих графических редакторов имеет общие инструменты обработки изображения (табл. 4.9).

Набор цветов экрана представляют собой *палитру изображения*. Как и у художника, она выполнена на экране в виде меню с набором красок. Но в отличие от палитры художника палитра графического редактора содержит еще образцы узоров, заложенные в меню редактора. Необходимо помнить, что «бумагой» или «холстом» служит сам экран, на котором вы рисуете цветом компьютерных инструментов: карандаша, фломастера, аэрозоли или кисти. Поэтому для экрана существуют понятия «цвет

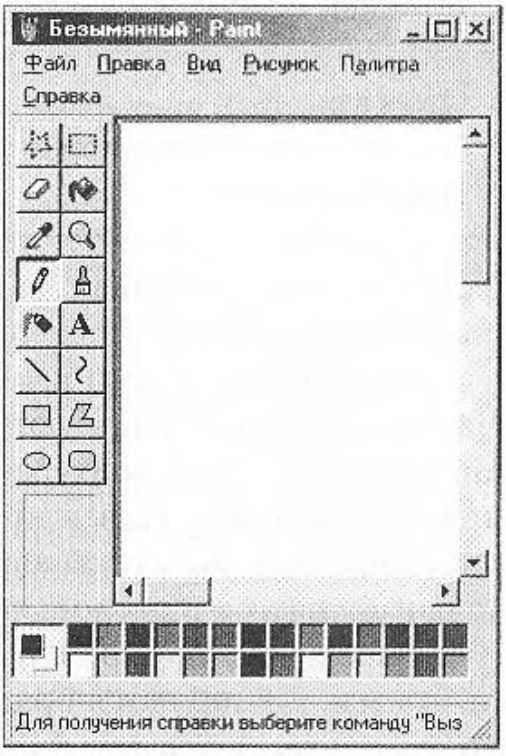


Рис. 4.9. Интерфейс программы Paint

Таблица 4.9

Инструмент	Описание
Карандаш, кисть	Проводит линию на экране — след от перемещения мыши при нажатой левой клавише
Ластик	Стирает на экране цветом, установленным в палитре
Валик или банка с краской	Выполняет заливку выбранной краской до контура рисунка
Круг-овал	Рисует круг или овал нужного размера
Квадрат-прямоугольник	Рисует квадрат или прямоугольник
Прямая линия	Проводит линию — линейку
Кривая линия	Проводит линию с искажением, как на лекало
Ножницы	Вырезает часть рисунка
Очистить	Очищает весь экран
Палитра красок	Показывает коробку с красками

фона» и «цвет рисунка» на этом фоне. Если вы закажите эти цвета одинаковыми, то не увидите рисунка, так как он сольется с фоном.

Цвет фигуры можно изменить — *залить* его краской до контура. Часто для этого используют пиктограмму инструмента в виде лейки. Но помните, если в контуре фигуры есть разрыв (хотя бы один незакрашенный пиксел), вся краска растечется из контура и испортит рисунок. Как узнать о наличии разрыва? Можно посмотреть подозрительный участок в режиме *лупа* и залатать прореху!

Бывает, что в порыве творчества мы берем не ту краску. В отличие от бумаги, где придется браться за дело заново, на компьютере можно *отменить ошибочное действие*, используя соответствующую услугу — **Отменить** в меню **Правка**. Обратите внимание, что эта услуга является частью интегрированной среды, т. е. применима в любых

редакторах, в том числе для работы с текстом, числовыми данными, при редактировании записей в электронных таблицах. Программная среда помнит несколько шагов, которые выполнялись вами в среде, и, отменяя действие, вы можете выполнить эту услугу многократно, вернувшись к тому моменту, с которого считаете возможным начать работу вновь.

Небольшую ошибку можно исправить с помощью ластика. Причем ластик на компьютере может приобретать цвет бумаги! Стираешь неверный рисунок, а на самом деле замазываешь его цветом фона.

Даже простой графический редактор содержит инструменты, позволяющие редактировать рисунок, управлять им: менять размер (масштабировать), деформировать (сжимать и растягивать), поворачивать, отражать.

Описание типовых команд графических редакторов приведено в табл. 4.10.

Таблица 4.10

Команда	Описание	Возможность
Меню «Правка»		
Отменить	Отмена ошибочного действия	Продвижение по шагам действий назад
Масштаб	Приближение изображения до его увеличения с видимыми пикселами	Коррекция изображения по пикселам
Показать сетку	Клетка пикселов на экране	Показ пустых и закрашенных пикселов, которые можно редактировать
Копировать	Копирование вырезанного фрагмента в промежуточную память	Формирование «буфера» (промежуточной временной области) памяти, в котором хранится фрагмент и из которого его можно восстанавливать неограниченное число раз, т.е. можно осуществить многократное копирование — размножение фрагмента
Вставить	Вставка на экран изображения из промежуточной памяти	

Окончание табл. 4.10

Команда	Описание	Возможность
Меню «Рисунок»		
Повернуть	Перемещение вырезанного фрагмента рисунка на угол, указанный в настройках	Вращение фрагмента, что позволяет путем многократного копирования с поворотом сформировать осесимметричные узоры из одного фрагмента
Отразить	Симметричное отображение вырезанного фрагмента	Отражение фрагмента относительно горизонтальной линии и вертикальной линии
Растянуть	Растяжение и сжатие по горизонтали и вертикали	Деформация рисунка
Наклонить	Наклон на градус, указанный в настройках	Потеря при наклоне фрагмента четкости изображения из-за искажения: растяжения вдоль наклона
Атрибуты рисунка	Установка размера «бумаги» для рисования	Экономия памяти, которую займет рисунок, путем выбора размера «бумаги»: больший размер «бумаги» требует большей памяти

Можно применять инструмент и к заранее указанной части рисунка. Для этого необходимо с помощью компьютерных ножниц (пунктирный прямоугольник или пунктирная ломаная) окружить мышью или компьютерным пером выбранный фрагмент рисунка, а затем применить к нему инструменты управления этим фрагментом. Вырезанный фрагмент расположен на «бумаге», которую можно сделать непрозрачной и прозрачной с помощью команды **Непрозрачный фон** в меню **Параметры**.

? Вопросы и задания

1. Перечислите общие инструменты графического редактора. Охарактеризуйте их.

2. Сравните традиционные инструменты художника с их аналогами в среде графического редактора.
3. Какая команда используется для отмены действия?
4. Выполните моделирование изображений снеговика как эмблемы новогоднего праздника с помощью инструментов графического редактора (кругов различных размеров и цвета).

П-Г Проектное задание

Подготовьте модель эмблемы вашего образовательного учреждения в графическом редакторе. Можете воспользоваться графическим планшетом для выполнения рисунков с помощью специального пера, а не мыши.

Используйте эмблему для проекта «Визитная карточка», для чего воспользуйтесь режимом вставки изображения в текстовый файл с визитной карточкой.

4.6. Графика в профессии

Графические редакторы позволяют компьютеризировать профессии художников-мультипликаторов, рекламных дизайнеров, художников — оформителей книг, альбомов, художников-модельеров, web-дизайнеров. Помогают они и в чертежной профессиональной деятельности. Примером служит компьютерная проектная среда, позволяющая оформлять планировку жилых и производственных помещений. Пусть ваш кабинет является рабочим помещением, в планировке которого учитывались санитарные нормы размещения оборудования, нормы пожаро- и электробезопасности. Давайте с учетом этих норм рассчитаем количество столов с оборудованием в вашем помещении и спланируем их расстановку согласно правилам техники безопасности. Перечислим такие правила для ИКТ-кабинета:

1) на одно автоматизированное рабочее место должно отводиться от 4 до 6 м^2;
2) задняя панель монитора должна быть обращена к поверхности стены;
3) столы должны выполнять роль электроограждения от проводки и розеток подключения компьютеров;
4) входная дверь должна иметь свободный доступ от всех столов кабинета в целях пожаробезопасности;
5) свет на доску должен падать слева;
6) преподаватель должен сидеть лицом или боком ко всем рабочим местам для полного обзора работающих за оборудованием.

Для создания чертежа можно использовать графический редактор Visio (http://microsoft.com/visualstudio/ru-ru/products/2010editions/express, http://technet.microsoft.com/ru-ru/office/ee236295 (рис. 4.10).

При работе с Visio пользователю доступны инструменты, отображенные в табл. 4.10.

Подробное описание средств программной среды дается в руководстве пользователя для программного продукта. Использование этих инструментов достаточно близко по способу применения в графических редакторах.

Особенностью подготовки документа в графическом редакторе является то, что он предназначен для вывода на чертежную бумагу: ватман или кальку. Для этого существуют специальные принтеры и плоттеры. Плоттеры позволяют получать чертежи высокого качества в автоматическом режиме, однако их подготовка ведется на компьютере в специальных графических средах профессионального назначения.

В последнее время широкое распространение получили графические инструменты фирмы *Adobe*, в частности Photoshop. В нем можно обрабатывать как уже имеющиеся изображения, так и полученные нами путем сканирования или извлеченные из карт памяти цифровых видео- или фотокамер (рис. 4.11).

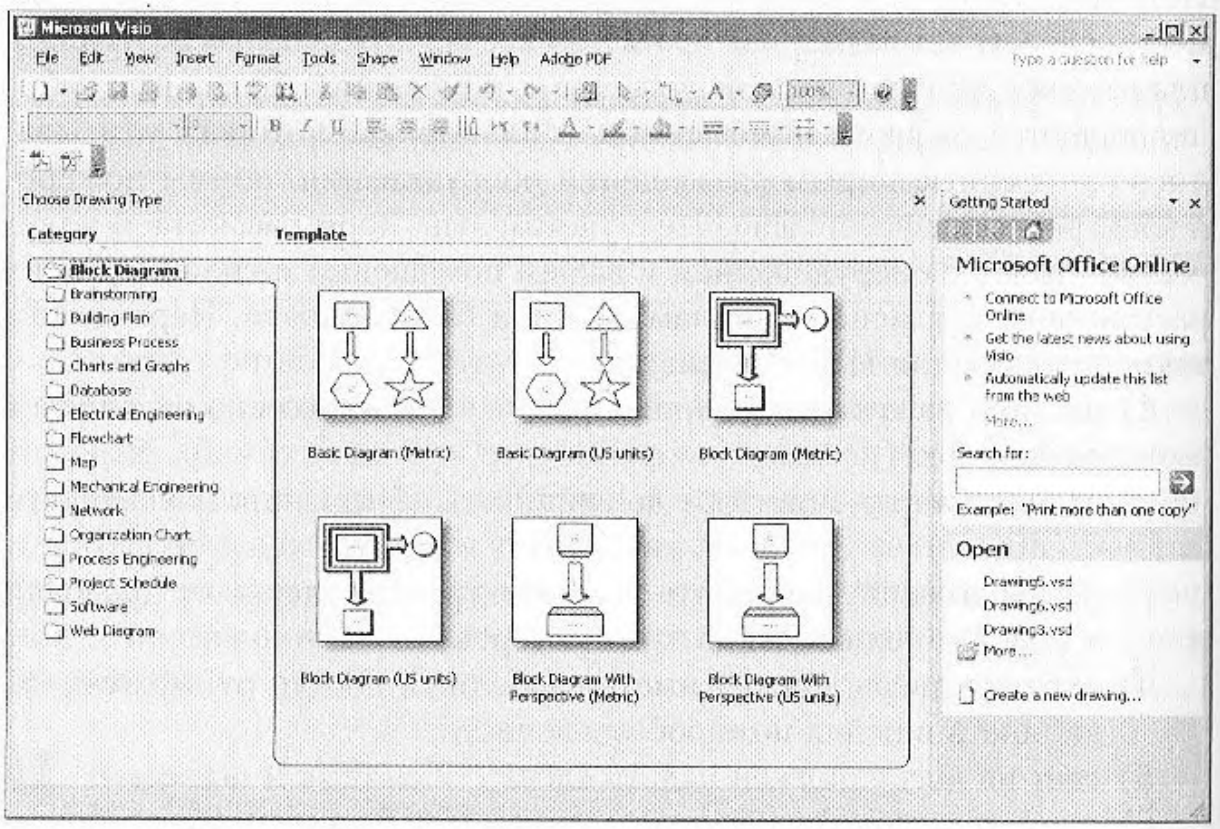


Рис. 4.10. Интерфейс программы MS Visio

Таблица 4.10

Номер	Назначение	Номер	Назначение
1	Создать рисунок	16	Уменьшить шрифт
2	Открыть рисунок	17	Толщина линии
3	Открыть набор примитивов	18	Тип линии
4	Сохранить рисунок	19	Тип концов линии
5	Печать	20	Скругление углов
6	Просмотр печати	21	Тень
8	Переход на предыдущую страницу	22	Привязка
7	Переход на следующую страницу	23	Приклеивание
9	Выравнивание примитивов	24	Увеличить масштаб (на экране)
10	Соединение	25	Уменьшить масштаб (на экране)
11	Вертикальное отражение	26	Нормальный режим редактирования
12	Горизонтальное отражение	27	Режим рисования примитивов
13	Поворот по часовой стрелке на 90°	28	Соединение примитивов
14	Поворот против часовой стрелки на 90°	29	Режим редактирования текста
15	Увеличить шрифт	30	Режим вращения примитивов



Рис. 4.11. Цифровая фотокамера

Сканирование осуществляется с помощью специального внешнего устройства — сканера и специальной программы (рис. 4.12). В системе Windows для этого предлагается программа MS Photo Editor. Для того чтобы осуществить сканирование, нужно поднять крышку сканера и положить лицом вниз рисунок или фотографию, которую надо отсканировать, затем запустить программу MS Photo Editor. В меню выбрать позицию **File** (файл), а в открывшемся подменю — команду **Scan Image**, после чего откроется специальное окно, в котором пока видно только изображение, сканированное ранее. Затем выполнить команду **Prescan** (предварительное сканирование), в результате чего на экране появится изображение, подлежащее сканированию (рис. 4.13).



Рис. 4.12. Сканер

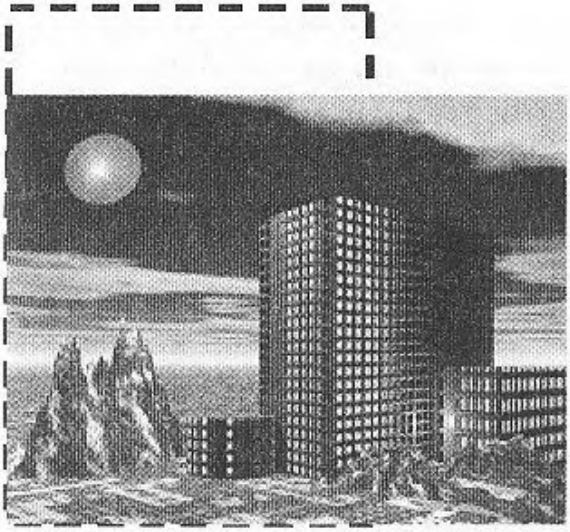


Рис. 4.13. Результат выполнения команды Prescan



Рис. 4.14. Выделение участка изображения

Рамка обозначает тот участок рисунка, который будет отсканирован с помощью нажатой левой кнопки мыши. Вы можете, регулируя размеры рамки, выбрать нужный участок (рис. 4.14). После чего следует нажать кнопку **Scan**. В результате начинается сканирование изображения, по завершении чего оно появится на экране (рис. 4.15).

Теперь это изображение можно сохранить на диске в одном из графических форматов, например BMP, JPG, TIFF.

Запустив программу Photoshop, через меню **Файл**, **Открыть** можно получить наш рисунок. Очевидно, что вызывать его нужно под тем именем, под которым вы сохранили его на диске. Теперь можно редактировать рисунок. Например, используя команды меню **Изображение**, **Повернуть**, **Зеркально по горизонтали**, можно получить симметричное по горизонтали изображение (рис. 4.16), а используя ко-

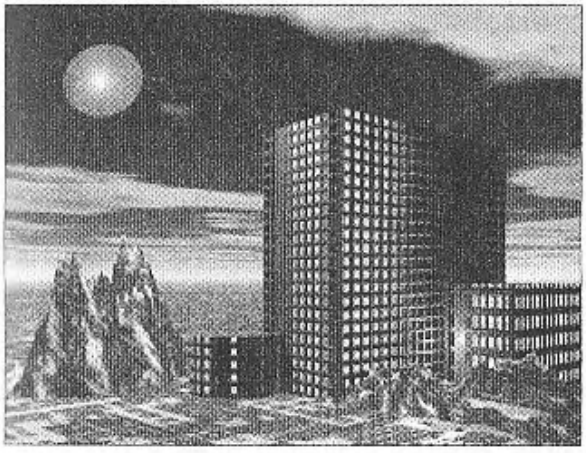


Рис. 4.15. Результат сканирования изображения

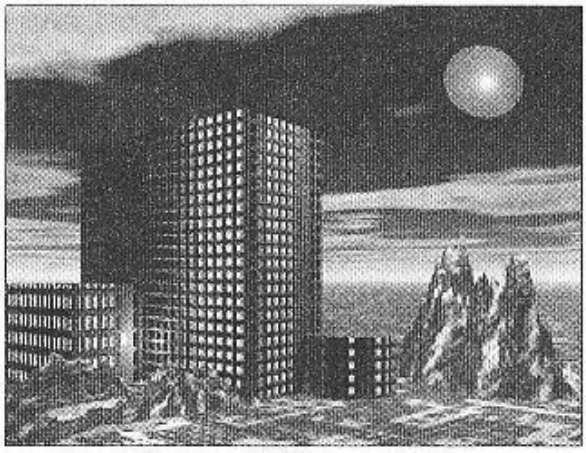


Рис. 4.16. Результат поворота изображения и зеркалирования его по горизонтали

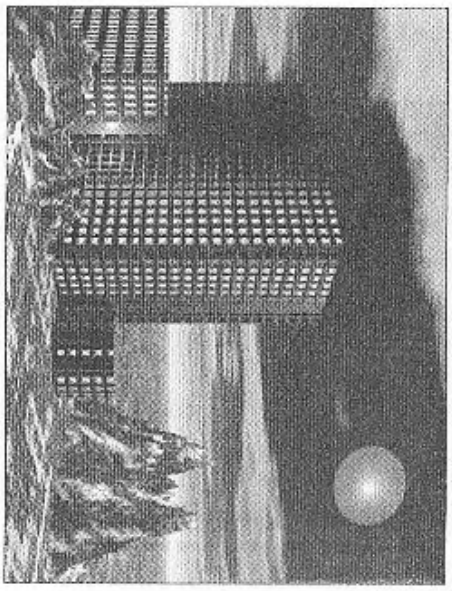


Рис. 4.17. Результат поворота изображения

манды меню **Изображение**, **Повернуть**, **Повернуть на 90° по часовой стрелке**, — перевернутое изображение (рис. 4.17).

Пользуясь командами меню, можно задать масштаб с помощью команд меню. Кроме того, размер рисунка можно изменить и традиционным способом — выделив его, установить курсор на границе рисунка до появления значка ↔, а затем при нажатой левой клавише мыши растягивать или уменьшать.

Современные инструментальные средства часто используют *библиотеки рисунков*, которые можно вставлять в текстовые, табличные и иные документы.

Например, в текстовом редакторе Word такая библиотека называется Clip Art и помимо рисунков содержит короткие видеоклипы, которые вставляются так же, как и рисунки, т. е. **Вставка**, **Рисунок**, **Картинки**. Вы можете создавать собственные коллекции изображений, поместив их в отдельный файл, тогда их вставка в документ будет происходить чуть иначе: **Вставка**, **Рисунок**, **Из файла**, после чего необходимо указать имя файла, в котором вы сохранили свои изображения. Очень часто коллекции изображений формируют по разделам «Наука», «Спорт», «Отдых», «Учеба», группируя там соответствующие рисунки. Это значительно облегчает поиск. Кроме того, огромные коллекции рисунков на самые разные темы можно найти в Интернете, для этого лучше всего воспользоваться поисковым сервером (Yandex, Rambler, Yahoo, Google и т. п.), задав соответствующие критерии поиска, например «Компьютерные картинки».

Некоторые примеры картинок из коллекции Clip Art, посвященные теме «Животные», представлены на рис. 4.18.

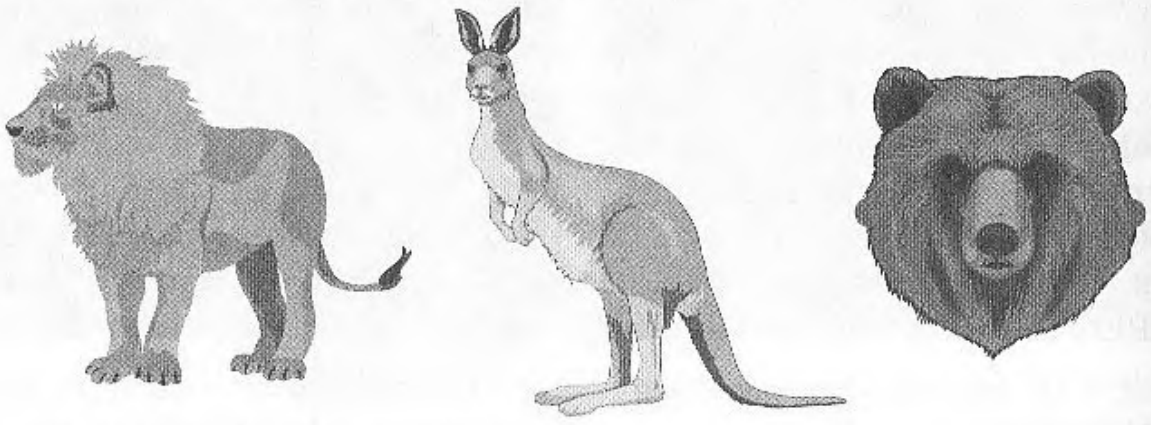


Рис. 4.18. Примеры изображений животных из коллекции Clip Art

? Вопросы и задания

1. Перечислите инструменты, позволяющие управлять рисунком в разных графических редакторах.
2. Какое цифровое оборудование необходимо использовать для создания цифровых фотографий, видеоклипов, копий рисунков?
3. Задание «Бабочка». Нарисуйте овал. Вырежьте его как фрагмент. Наклоните на 45°. Отразите относительно горизонтали. Сгруппируйте как крыло бабочки. Вырежьте его как фрагмент. Отразите относительно вертикали. Оформите цветом.
4. Задание «Обои». Нарисуйте геометрический узор, копируйте вдоль горизонтали.
5. Задание «Платок». Нарисуйте квадрат. Оформите его узором. Отразите вниз, а затем вправо. Из четырех деталей сформируйте платок.

П-Г Проектное задание

«Художник-модельер». Используя графический редактор, оформите в нем с помощью линий «манекены» и «оденьте» их, используя заготовки прямоугольника, круга, треугольника, трапеции. Наклоном, растяжением, сжатием, поворотом и отражением подгоните заготовку к деталям одежды, раскрасьте детали. Посмотрите модные стили на портале http://www.womenclub.ru/.

П-Т Проектное задание

Плакат «ИКТ-кабинет». Подготовьте в графическом редакторе планировку помещения, расстановку в нем рабочих столов с компьютерами и схему проводки электропитания и локальной сети к ним. Включите в плакат требования к рабочим местам с компьютером.

4.7. Видеомонтаж

Отсняв на цифровую видеокамеру материал, например о вашем учебном заведении, можно, подключив видеокамеру к компьютеру, оснащенному специальной платой, «перекачать» этот материал в видеофайл с расширением .avi и потом просмотреть его. Это достаточно несложно и делается с помощью относительно простых программ. Однако для создания действительно интересного видеожурнала необходимо произвести еще и видеомонтаж. Из несложных в использовании программ можно назвать VideoWave, Windows Movie Maker. Есть программы помощнее: Ulead Media Studio; M2-Edit Professional; профессиональный редактор MPEG-2, который позволяет менять поток, редактировать, производить поиск, перепаковку видеопакетов MPEG-2 и MPEG-1, быстрый просмотр и т. д. Если заниматься видеомонтажом серьезно, то лучше остановиться на Adobe Premiere. Эта программа, появившаяся в 1991 г., стала первопроходцем в области нелинейного видеомонтажа на настольных системах.

Для краткого ознакомления с программами обработки видеофайлов рассмотрим Windows Movie Maker. Открыть эту программу можно из меню **Пуск**, **Windows Movie Maker**. Главное окно программы состоит из нескольких областей: панель операций с фильмами, панель содержимого файлов, панель монитора (для просмотра клипов и всего фильма) и панель редактирования изображений (совмещающая панели раскадровки и шкалы времени) (рис. 4.19).

Практически вся работа по созданию и монтажу проекта производится в области редактирования изображений на двух переключающихся между собой панелях: «Отображение раскадровки» и «Отображение шкалы времени». *Раскадровка* используется для работы с последовательностью кадров (клипов). Кроме того, на панели раскадровки отображаются все примененные *видеоэффекты* (рис. 4.20).

Шкала времени предназначена в первую очередь для работы с временны́ми параметрами клипов. К тому же на этой панели отображается содержимое добавленных в проект аудиофайлов (рис. 4.21).

AVI — формат несжатого видео, создаваемый при оцифровке изображения. Это наиболее ресурсоемкий формат, но при этом при оцифровке в него потеря данных минимальна. Поэтому он предоставляет больше возможностей для редактирования, накладки эффектов и любой другой обработки файлов. Однако следует учитывать, что в сред-

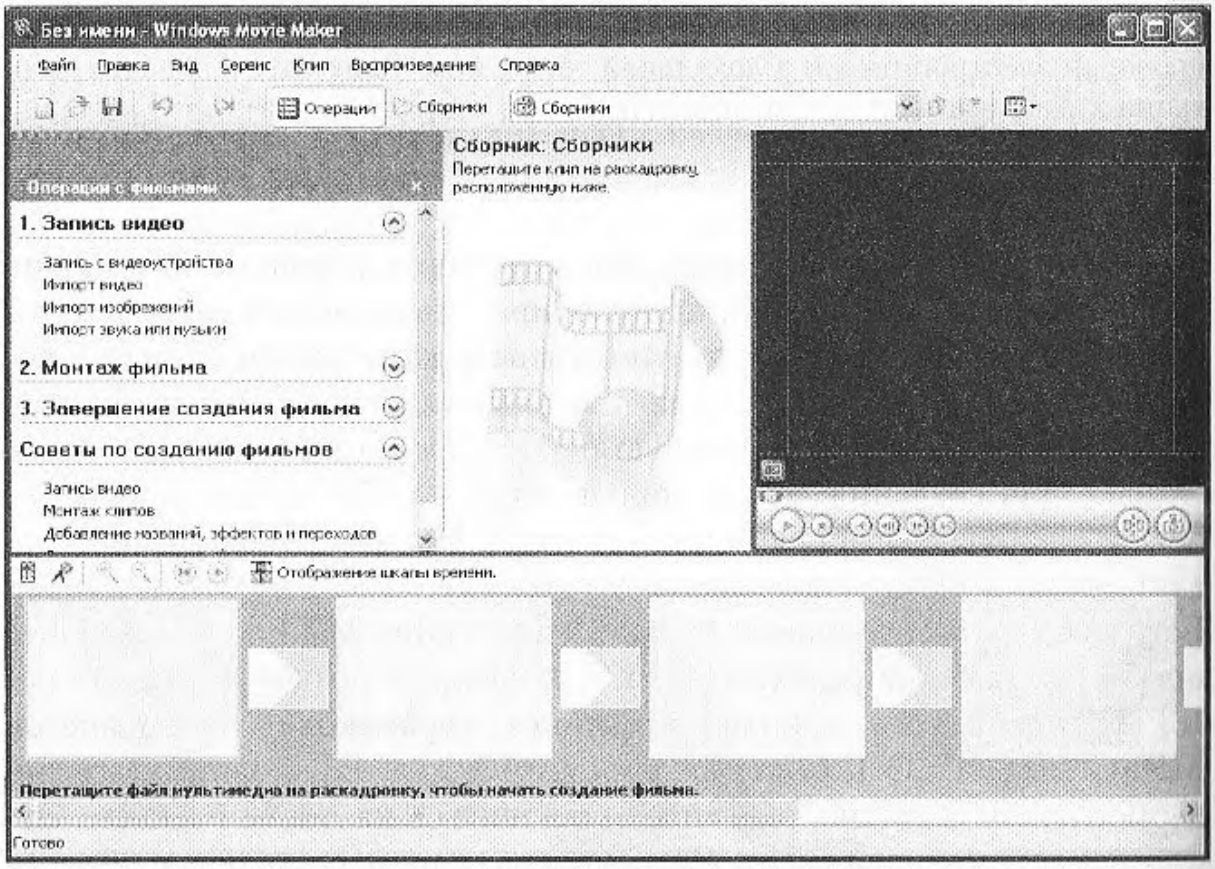


Рис. 4.19. Интерфейс Windows Movie Maker

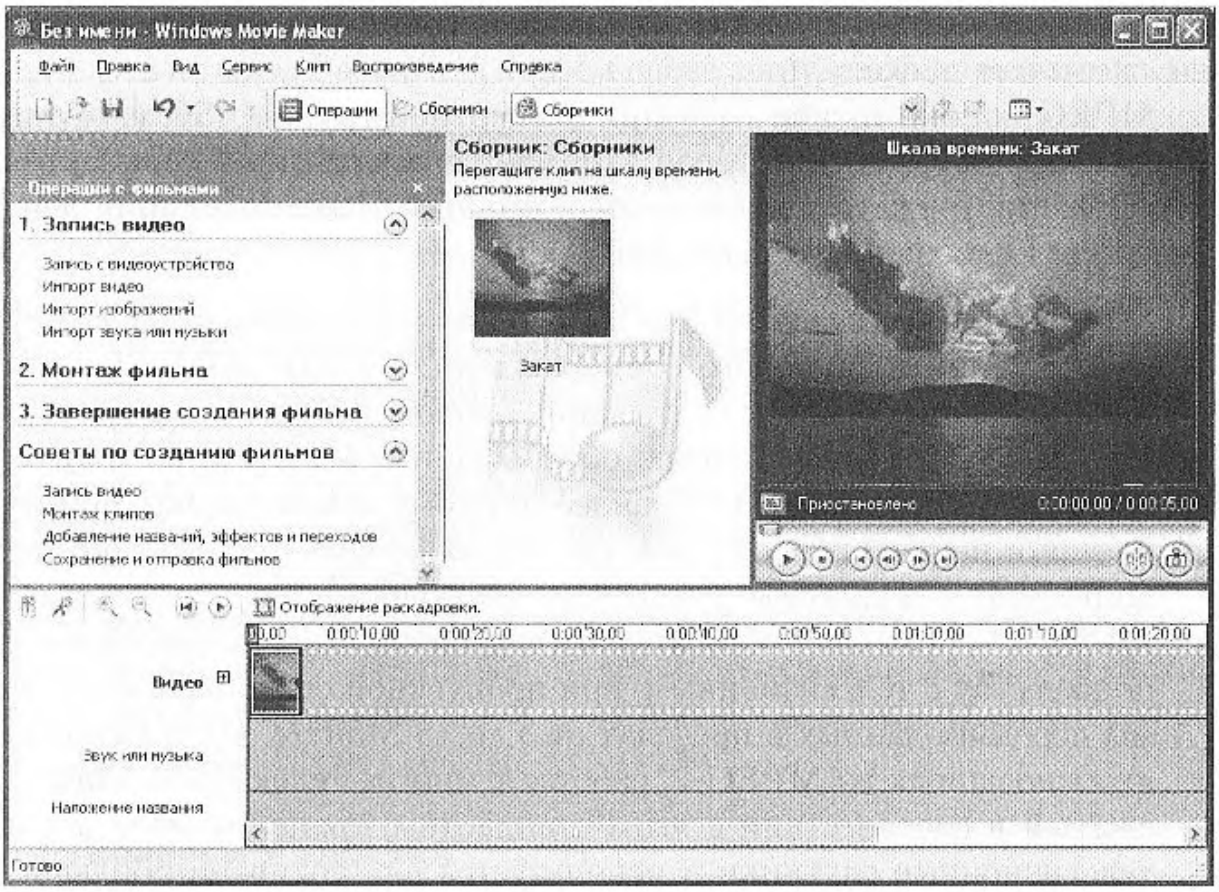


Рис. 4.20. Раскадровка

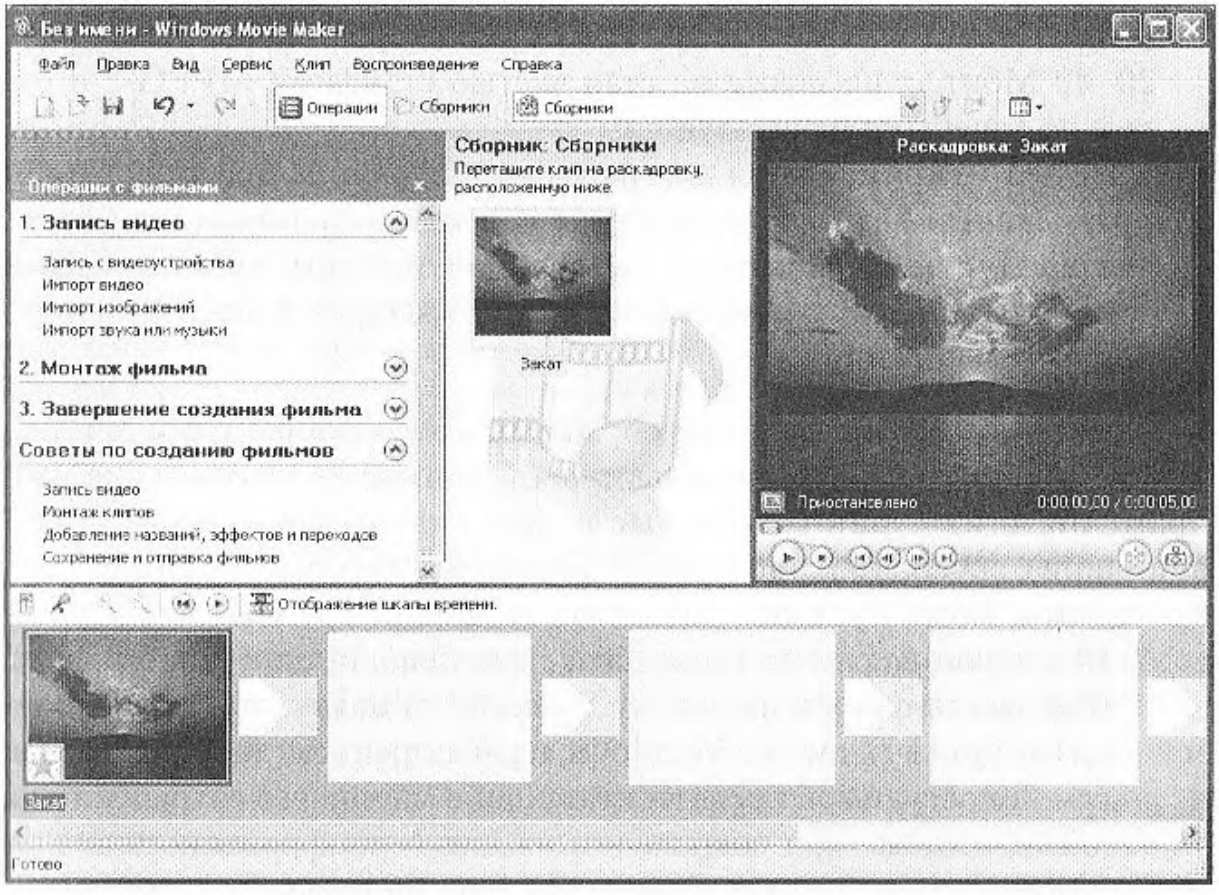


Рис. 4.21. Шкала времени

нем одна секунда цифруемого изображения занимает 1,5—2,0 МБайт на «жестком» диске.

MPEG — сокращение названия экспертной группы ISO (Moving Picture Expert Group), которая занимается разработкой стандартов кодирования и сжатия видео- и аудиоданных. На сегодняшний день известны следующие форматы MPEG:

- MPEG-1 — предназначен для записи синхронизированных видеоизображений и звукового сопровождения на CD-ROM с учетом максимальной скорости считывания около 1,5 Мбит/с. Качественные параметры видеоданных, обработанных MPEG-1, во многом аналогичны обычному VHS-видео, поэтому такой формат применяется в первую очередь там, где неудобно или непрактично использовать стандартные аналоговые видеоносители;
- MPEG-2 — позволяет обрабатывать видеоизображение, соизмеримое по качеству с телевизионным, при пропускной способности системы передачи данных в пределах от 3 до 15 Мбит/с. На технологиях, основанных на MPEG-2, работают многие телеканалы; сигнал, сжатый в соответствии с этим стандартом, транслируется через телевизионные спутники и используется для архивации больших объемов видеоматериала;
- MPEG-3 — используется в системах телевидения высокой четкости (High-Defenition Television — HDTV) со скоростью потока данных 20–40 Мбит/с; но позже он стал частью стандарта MPEG-2 и отдельно теперь не используется;
- MPEG-4 — применяется для работы с цифровым представлением медиаданных в трех областях: интерактивном мультимедиа (включая продукты, распространяемые на оптических дисках и через сеть), графических приложениях (синтетического контента) и цифровом телевидении.

Поскольку многие средства обработки видеоизображений используют видеофайлы, файлы компьютерной анимации, статические изображения и их серии (отдельные кадры видео- или компьютерной анимации), звуковые клипы, то все форматы, с которыми работают эти программы, можно условно разделить на четыре группы.

1. *Форматы файлов фильмов* (.avi, .mov, .flm, .dv, .mpeg, .mpe, .mpg). Файлы фильмов в форматах Video for Windows с расширением .avi и Quick Time for Windows с расширением .mov создаются самой программой после монтажа фильма при его сохранении на диске, а также при сохранении на диске оцифрованного аналогового видео. Формат FilmStrip (.flm) применяется для преобразования видео в серию отдельных несжатых кадров, которые

потом могут быть импортированы и обработаны графическим редактором Photoshop, а затем снова загружены в программу. Это очень удобно, когда нужно применить ко всем кадрам видео какой-либо эффект из набора Photoshop.

2. *Файлы статических изображений* (Photoshop — .psd, Windows Bitmap — .bmp,. .rle, .dib; Windows Metafile — .wmf, .tif, .pcx; Targa — .tga, .vda, .icb, .vst; Jpeg — jpg, .jpe, .jfif и др.). Такие файлы могут создаваться в любом графическом редакторе, после чего их можно сконвертировать в любой формат, поддерживаемый программой.
3. *Звуковые файлы* (.aif, .wav).
4. *Файлы серий статических изображений и компьютерной анимации.* Файлы анимации создаются специальными программами компьютерной графики (.flc, .fli). Кроме того, некоторые программы могут генерировать серии нумерованных кадров статических изображений, которые при последовательном воспроизведении создают эффект анимации.

? Вопросы и задания

1. Для каких целей используется программа Adobe Premiere?
2. Опишите интерфейс программы Windows Movie Maker.
3. Перечислите элементы окна монтажа.
4. Опишите видеоформаты.

П Проектное задание

Создайте видеожурнал об информационной среде вашего учебного заведения, в состав которой могут входить ИКТ-кабинет, предметные компьютеризированные кабинеты, медиазал, студия аудио- и видеомонтажа, серверная, локальная сеть учреждения, автоматизированное рабочее место (АРМ) администратора, радиоузел, телестудия, цифровая естественно-научная лаборатория, лаборатория робототехники, АРМ библиотекаря, медиатека и пр.

4.8. Автоматизированное проектирование

Широкое внедрение компьютеров в профессиональную сферу в условиях научно-технического прогресса обеспечивает рост производительности труда. Если производительность труда в сфере производства с начала XXI в. возросла в сотни раз, то в области проектирования только в 1,5—2 раза. Это обусловливает большие сроки проектирования новых объектов, что не отвечает потребностям развития экономики. Одним из направлений решения этой проблемы

является создание и развитие систем автоматизированного проектирования.

В настоящее время продолжительность проектирования большинства сложных объектов превышает разумные пределы, определяемые сроками морального износа, утратой конкурентоспособности изделий и др. Поскольку сложность объектов будет возрастать, а время проектирования должно сокращаться, можно сделать вывод, что единственной альтернативой неавтоматизированному проектированию может быть широкое использование САПР.

Системы автоматизированного проектирования должны:

1) автоматически выявлять оптимальные решения;
2) использовать единые информационные системы и БД для всех составных частей САПР;
3) предоставлять проектировщику возможность вносить коррективы в первоначально принятые системой решения;
4) автоматически проверять совместимость решений, принимаемых по разным частям и элементам проектируемого объекта, и осуществлять (когда это возможно) корректировку несовместимых решений без учета проектировщика;
5) автоматически выдавать любые промежуточные результаты по запросу проектировщика;
6) давать возможность изолированного решения отдельных задач по задаваемым проектировщиком исходным данным;
7) накапливать опыт проектирования, обеспечивать возможность совершенствования и развития системы без ее коренной переделки;
8) выдавать по запросу любые сведения, хранящиеся в банке данных системы.

Все эти требования могут быть сведены к двум важнейшим качествам системы: *информированность* и *адекватность*. Именно они практически полностью определяют успех внедрения и эксплуатации САПР.

В настоящее время существует множество программных средств, на базе которых реализованы САПР: AutoCAD, AutoCAD LT, AutoSketch R7 и др.

Существуют САПР, специализирующиеся в различных отраслях. Например, наиболее применяемыми в машиностроении являются:

- Mechanical Desktop R6 Power Pack;
- Autodesk Inventor R5;
- MechSoft для Autodesk Inventor;
- Cadmech;

- Cadmech Desktop;
- AVS;
- Techcard;
- WinStructure 3D.

В архитектуре и строительстве широко распространены следующие САПР:

- Autodesk Architectural Desktop;
- Autodesk Building Systems;
- «Маэстро»;
- Rebis AutoPLANT Structural;
- LCAD и т.д.

Существуют специальные средства обеспечения дизайна, например 3D Studio MAX, Autodesk VIZ, специальные средства так называемой векторизации чертежей, например GTXRaster Tools и пр.

Попробуем понять, как строится простейший чертеж с помощью достаточно простой системы «Кедр» (http://www.kedrwin.com/download.htm). Выполним чертеж ИКТ-кабинета.

Новый чертеж можно создать, осуществив последовательно действия, приведенные ниже.

1. Запустить программу Kedrwin из директории, где она записана. В меню **Файл** выбрать пункт **Новый чертеж** (рис. 4.22).

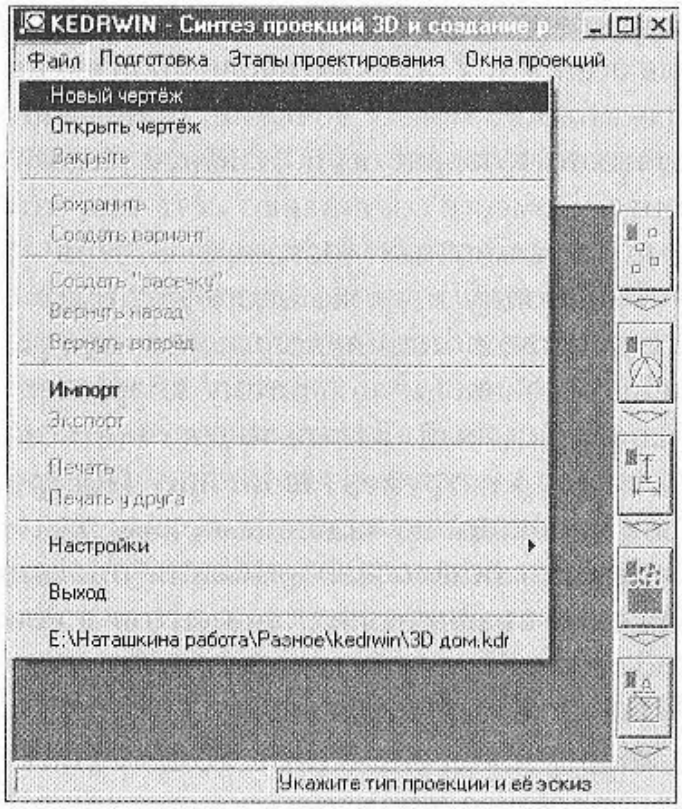


Рис. 4.22. Создание чертежа в системе проектирования Kedrwin

2. Определить проекцию объекта проектирования. Каждая проекция имеет свое название. Просто «Вид» — самое общее название и назначение проекции. Если для текущей проекции подходит одно из стандартных названий, то его и надо выбрать. Если у вас есть «оцифрованный» рисунок или фотография проектируемого объекта, то можно использовать его в качестве фона окна проекции объекта. На такой фон легче наносить изображение проекции. Такой фон называется *файлом эскиза проекции*. Если эскиза нет, то проекция «Без эскиза». Далее необходимо выполнить заготовку окна проекции — «растянуть» ее в окне, указать опорные точки изображения проекции. Если эскиза нет, то можно просто указать в любом месте хотя бы одну точку.

3. На панели инструментов нажать на кнопку **Ввод графических элементов проекции** (название кнопки отображается при наведении на нее курсора мыши). Вы перейдете на этап ввода графических элементов проекции. Используя имеющиеся опорные точки или создавая их при вводе графических элементов («зажатая» клавиша **Ctrl** при указании очередной точки элемента), нарисовать эскизное изображение проекции объекта. «Рисование» состоит в составлении изображения из графических элементов, которые устанавливаются на опорные точки.

4. Нажав на панели инструментов на кнопку **Ввод размерной сети проекции**, перейти на этап ввода размерной сети проекции. Размеры наносятся на сформированные вами опорные точки. На одну опорную линию может быть «нанизано» несколько опорных точек. Размеры направляются вдоль осей *X* и *Y*, а также вдоль производных осей. Размер, по сути, задает расстояние между точками, привязанными к опорным линиям вдоль соответствующих осей. Каждый размер вводится индивидуально, но они все вместе составляют сеть так, чтобы на проекции не было автономных размерных «островков». Под «соприкасанием» понимается то, что на одну и ту же опорную точку опирается более одного размера. Если точка была использована для размеров, которые составляют линейно независимую систему координат, то точка изменяет свой цвет на фиолетовый. Белых точек остаться не должно.

5. Нажать на панели инструментов кнопку **Построение истинного изображения всех проекций по заданным размерам** — будет запущен процесс построения истинного изображения проекции, которое состоит в расположении графических элементов в соответствии с заданной размерной сетью. Если размерная сеть была задана без ошибок, то построение успешно завершится.

6. Нажать на кнопку **Ввод элементов оформления чертежа**, перейдя тем самым на этап ввода элементов оформления. Указанные вами размеры будут автоматически размещены на проекции. Вы можете их

передвинуть, добавить им надписи, указать выноски, сечения, разрезы, текстовые надписи обозначения шероховатостей и определить области штриховки. Одним словом, оформить отдельно каждую проекцию.

7. Нажать на кнопку **Окончательное размещение всех проекций объекта на чертеже**, т.е. перейти на этап окончательного размещения всех проекций объекта на чертеже. Программа сама автоматически выберет формат чертежа и разместит на нем проекции.

8. Для заполнения штампа чертежа войти в пункт меню **Подготовка**, **Основная надпись**, **Да**. Заполнить поля основной надписи и закрыть диалоговое окно.

Программой предусмотрена отправка результата по почте на указанный в программе адрес (пункт меню **Файл**, **Настройки**, **Адрес сервера**), что позволяет организовать работу проектировщиков проекта в удаленном режиме.

На панели инструментов предусмотрена услуга «Подготовка вывода на принтер», позволяющая выбрать принтер, ориентацию его листов и получить распечатку чертежа.

П-Т Проектное задание

Выполните задание, рассмотренное в параграфе, и используйте его как фрагмент для курсовой работы в соответствии с вашей профессиональной подготовкой.

Например, подготовьте чертеж планировки помещения с установленным АРМ по профессиям.

4.9. Технология обработки звуковой информации

Рассмотрим возможности музыкального редактора. Есть несколько типов музыкальных редакторов: *нотный* (Cakewalk Overture, Denemo, Finale, Sibelius), который позволяет формировать нотный текст и проигрывать его, и *звуковой* (например, MIDI-редактор TuxGuitar), который обеспечивает прямое редактирование звука различными средствами. Они различаются по принципу работы и уровню сложности. Музыкальный редактор, как и другие виды редакторов, обрабатывает информацию — выполняет набор, исправление, сохранение и вывод ее на внешние устройства. Но для текста есть клавиатура, а для графики — мышь и компьютерное перо, позволяющие вводить информацию. А как же создавать нотный текст или звуковой файл? Об этом любая программа-редактор расскажет в режиме «Помощь». Для работы со звуком понадобятся внешние устройства — музыкальная клавиатура, динамики, микрофон. Чтобы грамотно работать с нотным текстом,

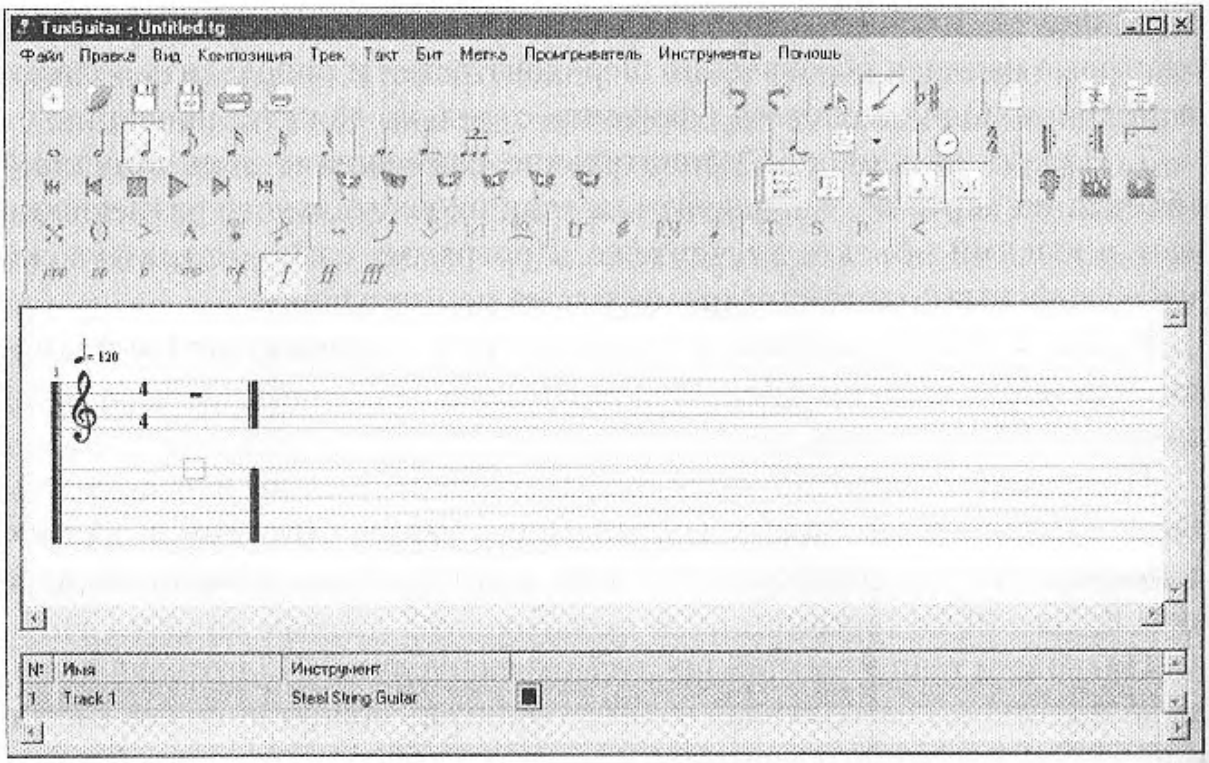


Рис. 4.23. Интерфейс музыкального редактора TuxGuitar

нужно знать нотную грамоту: нотный стан, звукоряд, длительность ноты, тон, полутон, диез, бемоль, бекар, пауза, длительность паузы.

Сила звука может быть различной. Громкое исполнение в музыке называют *forte*, тихое — *piano*, совсем тихое — *pianissimo*. Эти слова пришли к нам из итальянского языка.

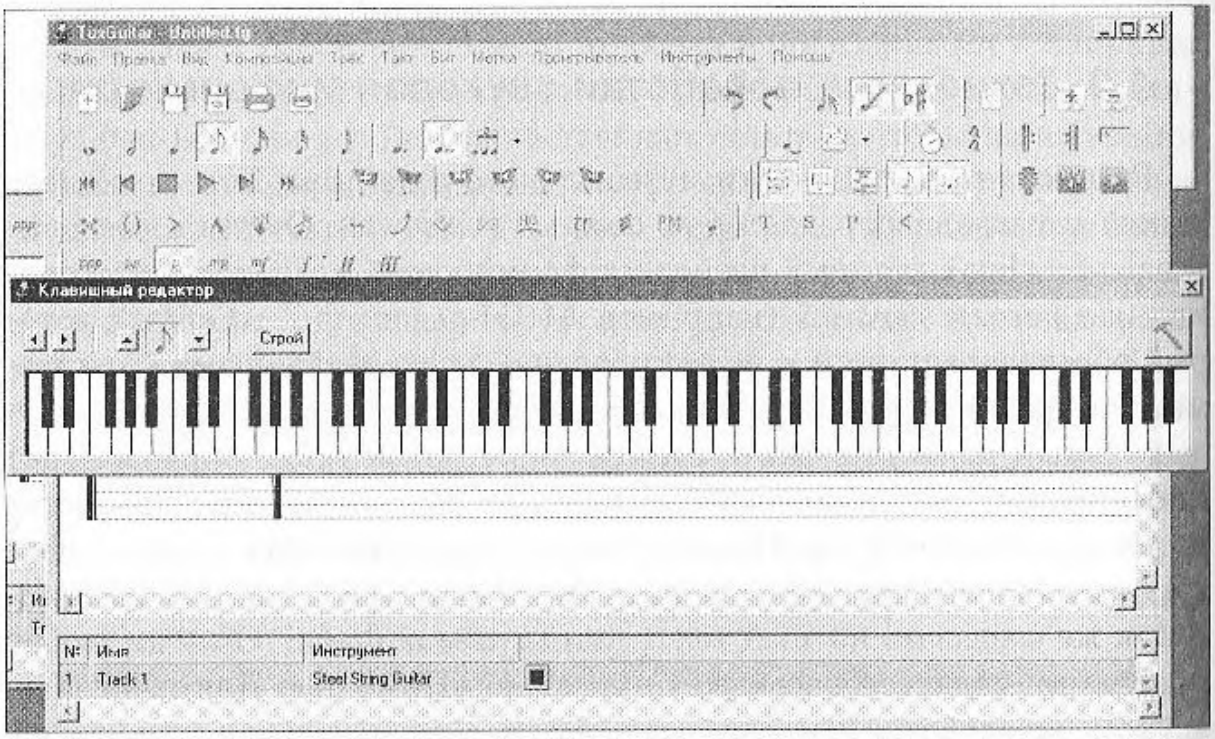


Рис. 4.24. Панель «Клавишный редактор» музыкального редактора TuxGuitar

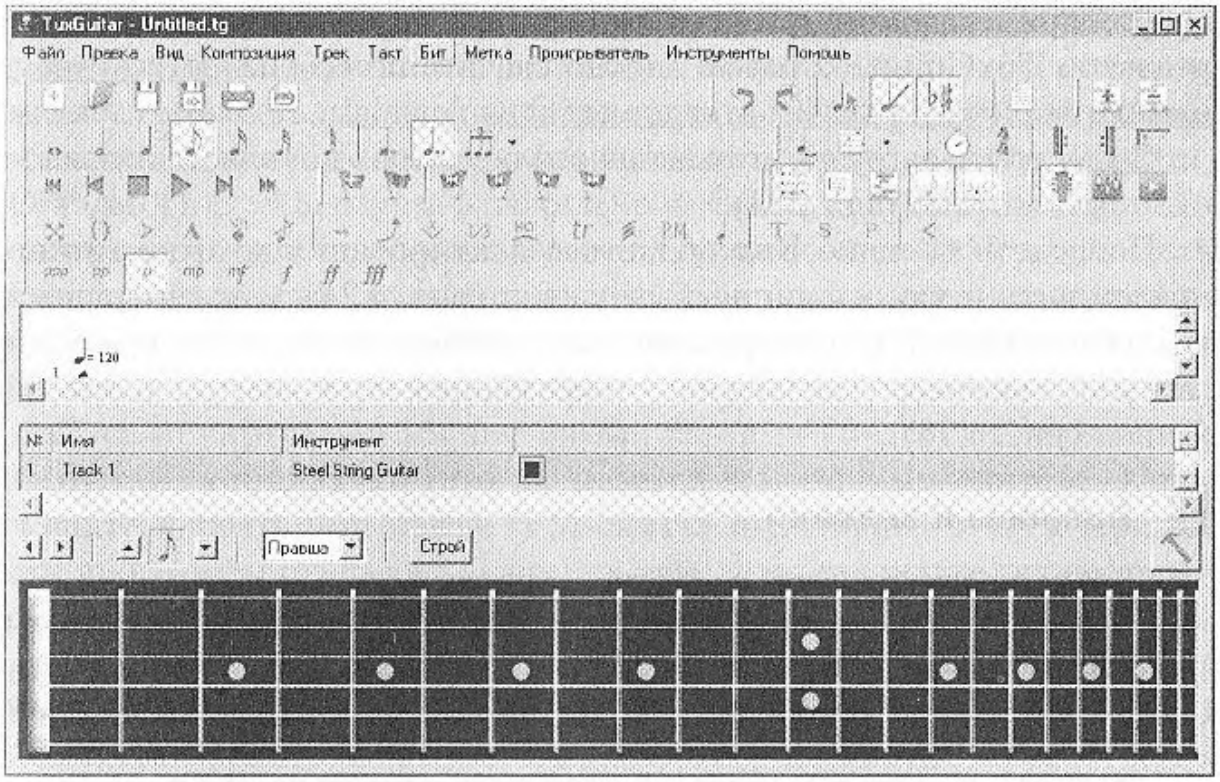


Рис. 4.25. Панель «Гриф» музыкального редактора TuxGuitar

Все элементы нотной грамоты описаны в меню музыкального редактора. Пользуясь ими, можно набрать нотный текст на компьютере. Как же можно прослушать набранный нотный текст? Для этого в компьютере есть устройство вывода звука — *динамик*.

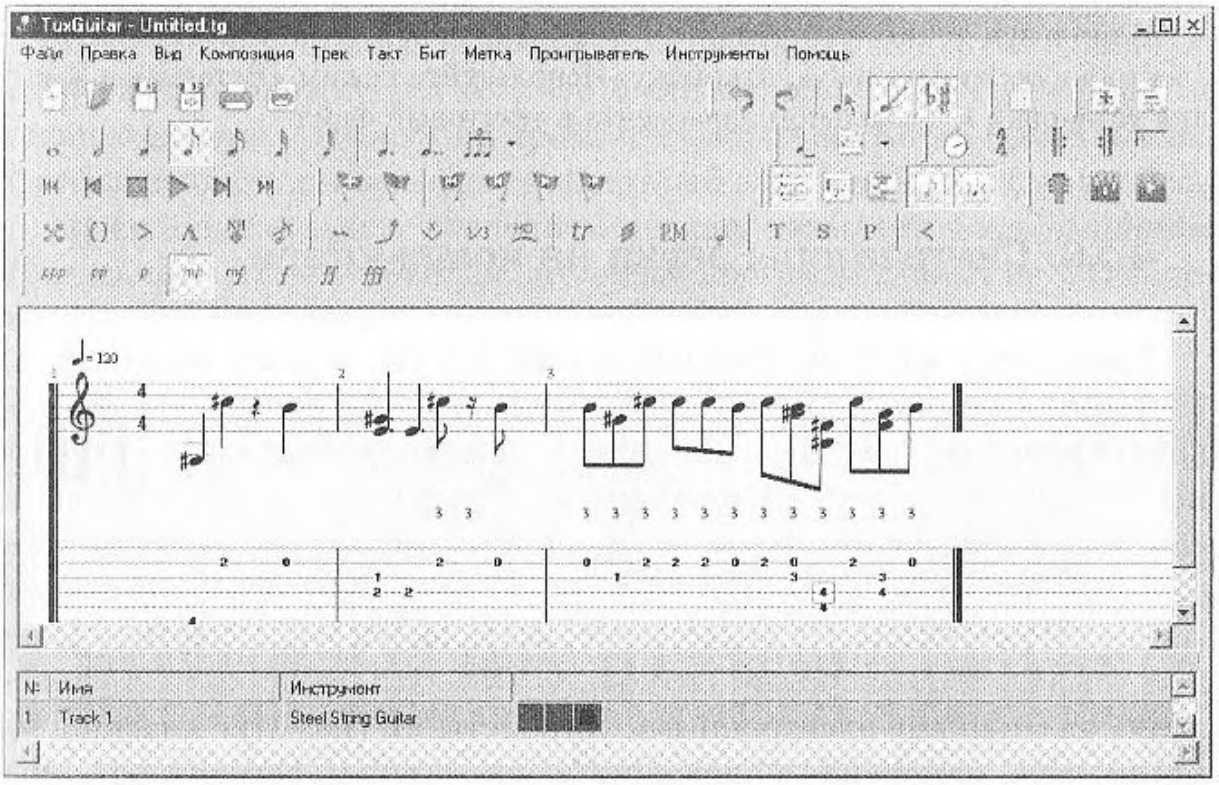


Рис. 4.26. Создание мелодии в музыкальном редакторе

Для примера рассмотрим свободно распространяемый музыкальный редактор TuxGuitar, который можно скачать по ссылке http://www.izone.ru/multimedia/utilities/tuxguitar-download.htm.

Сразу после загрузки музыкального редактора появляется главное окно программы (рис. 4.23).

Посредством меню **Вид** область редактирования музыки можно представить и как клавишный редактор (рис. 4.24), и как гитарный гриф (рис. 4.25). В редакторе присутствуют возможности для создания мелодии с помощью как клавиш, отображенных на экране, так и нотной записи (рис. 4.26).

? Вопросы и задания

1. Что такое музыкальный и нотный редакторы?
2. Перечислите основные цифровые устройства, необходимые для работы с музыкальным редактором.
3. Самостоятельно изучите музыкальный редактор. Прослушайте мелодии из компьютерной музыкальной библиотеки. Составьте конспект основных команд работы с нотным текстом в музыкальном редакторе. Воспользуйтесь помощью и выполните набор нотного текста небольшого произведения из сборника, прослушайте его.
4. Оформите музыкальное сопровождение для презентации своего образовательного учреждения.

П-Г Проектное задание

Используя текстовый редактор и музыкальный редактор, подготовьте электронную музыкальную открытку. Используйте режим «Вставка» в Word для установки музыкального фрагмента в текстовый файл.

4.10. Синтезаторы звука на компьютере

Компьютер на столе вмещает в себя целую *студию звукозаписи*. С чего же начинать запись песни? Ну конечно, надо сначала подготовить основу произведения. Для этого лучше всего подойдет виртуальная студия (типа Cakewalk или Cubase). С ее помощью можно выбрать (вырезать) фрагменты из песен (куплет, припев, проигрыш), установить порядок следования фрагментов и определить, сколько это займет тактов. Можно записать первоначальные аранжировки (партию ударных, баса) в MIDI-дорожки (треки). *MIDI* — это набор команд управления различными электронными устройствами (нажатие клавиши на клавиатуре, изменение положения какого-нибудь регулятора). Эти команды и записываются в MIDI-дорожки. А при воспроизведении

команды передаются через MIDI-кабель, соединяющий компьютер с синтезатором. *Синтезатор* выполняет набор команд — издает звуки, меняет громкость. Дальше вы будете делать аранжировку. Синтез звука можно поручить программам-синтезаторам (Generator, Reaktor, Orangator). В этом случае желательно иметь звуковую карту с возможностью загрузки подобранных вами звуков (семплов). У карты есть внутренняя память, куда помещаются с помощью специальных программ эти звуки, и вы можете использовать их для проигрывания.

Есть и другие варианты. Многие программные синтезаторы являются сами виртуальными звуковыми картами (Reality), и можно как загружать в них звуки (это называется семплерным синтезом), так и синтезировать ими новые (FM-синтез, аддитивный и др.). Правда, эти программы отнимают значительную долю мощности процессора.

Кроме того, можно использовать внешние синтезаторы (которые располагаются вне корпуса компьютера) — это знакомые вам клавишные инструменты и звуковые модули (коробочки с MIDI-входами (выходами), синтезаторы без клавиатуры (рис. 4.27)).

После того как будет готова фонограмма без голоса (минус), вы можете приступать к записи вокала. В той же виртуальной студии (Cubase, Cakewalk) наряду с инструментальными партиями можно записать несколько аудиодорожек (партий) вокала. На аудиодорожки можно записать еще и партии акустических инструментов. После этого нужно отрегулировать звучание голоса (тембр, придать объем с помощью ревербератора и т. д.), настроить уровень каждого инструмента, обработать его нужными эффектами, записать инструментальную стереоаудиодорожку (из тех, что были в формате MIDI), провести окончательную подгонку настроек всех дорожек. Когда песня готова, виртуальная студия создает стандартный звуковой файл (.wav). Весь процесс, описанный в последнем абзаце, называется *мастерингом* (сведением). Из wav-файла с помощью программ записи на CD можно создать аудиодиск.

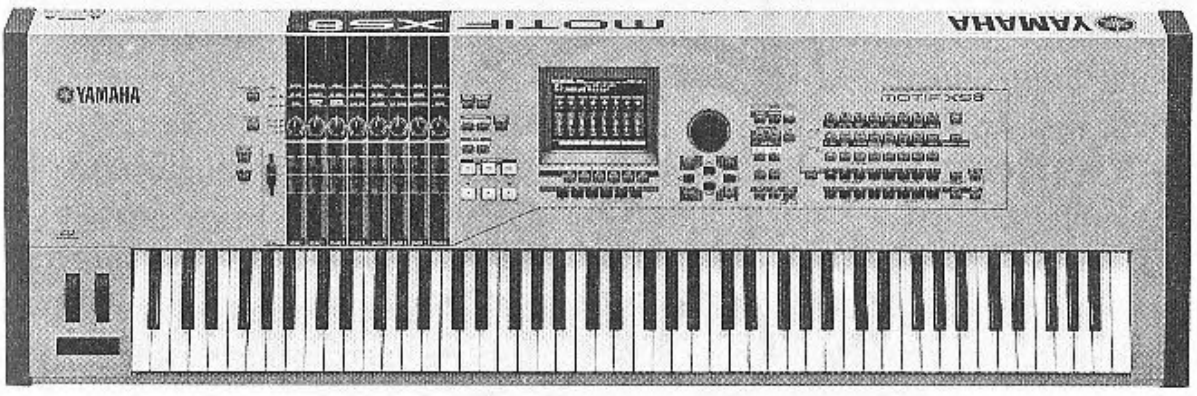


Рис. 4.27. Синтезатор Yamaha Motif

Те музыканты, которые не имеют дела с синтезаторами (например, электрогитаристы, барабанщики), могут пользоваться программами многоканальной аудиозаписи (Saw, Vegas Pro), которые отличаются от виртуальных студий, рассмотренных ранее, только отсутствием MIDI-дорожек. Для аудиозаписи они, в принципе, не нужны, хотя по MIDI могут управляться самые различные устройства, вплоть до гитарных и вокальных процессоров. В остальном же процесс записи «несинтезаторной» музыки ничем не отличается от записи электронной.

Чтобы иметь возможность играть и слушать музыку на компьютере, необходимо цифровое оборудование — компьютерные *звуковые колонки*, которые подключаются к *звуковой карте* (аудиокарте), находящейся внутри системного блока, к гнезду «Speakers» (громкоговорители), а если такового нет, то к «Out» (выход). Такой разъем (гнездо и штекер) называется «мини-джек». Для того чтобы компьютер мог записывать то, что вы скажете, напоете или сыграете, например на акустической гитаре, надо подключить микрофон (гнездо звуковой карты «Mic»). К линейному входу аудиокарты (гнездо «Line») можно подключить аудиовыход магнитофона, CD-проигрывателя, клавишного синтезатора (Korg M1, Roland D-50, Yamaha PSR 510). Если вы умеете или желаете исполнять ваши произведения на фортепианной клавиатуре, то имеется возможность подключить к звуковой карте MIDI-клавиатуру (прототип клавиатуры фортепьяно) — к разъему «MIDI-Joystic» с помощью специального MIDI-кабеля (рис. 4.28).

Вверху показана задняя стенка компьютера (выходы звуковой платы), а внизу — внешний синтезатор, т. е. вне корпуса компьютера (клавишный инструмент Korg), кабель с двумя пятиштырьковыми концами (DIN5) и одним 15-штырьковым — *MIDI-кабель*. Выходной конец кабеля («Out») надо подключить к MIDI-входу синтезатора



Рис. 4.28. Схема подсоединения аппаратуры

MIDI In (сюда побегут MIDI OUT ноты из компьютера, и «синтез» зазвучит), а входной («In») — к разъему синтезатора (здесь будут идти ноты от клавиатуры к машине). Сама MIDI-клавиатура, конечно, не может воспроизводить звуки. Она только выдает компьютеру номера нот (MIDI-команды), которые вы играете, а он заставляет синтезатор (генератор различных музыкальных звуков), имеющийся у звуковой карты, играть те ноты, которые вы нажали. Учтите, чтобы вы смогли играть на своей звуковой карте с помощью MIDI-клавиатуры, нужна еще и программа, например «Виртуальный MIDI-кабель», которая свяжет MIDI-вход карты с ее синтезатором.

Если вы хотите использовать внешний синтезатор (например, имеющийся у вас электронный клавишный инструмент), то с помощью второго разъема того же MIDI-кабеля вы должны подключить к нему MIDI-выход аудиокарты. В качестве MIDI-клавиатуры можно использовать клавишный синтезатор. Тогда к нему подключаются оба конца MIDI-кабеля (см. рис. 4.28).

Рассмотрим средства Windows для работы со звуком. При этом необходимо выполнить последовательность действий, приведенную далее:

1) нажать кнопку **Пуск** (в левом нижнем углу экрана), выбрать в появившемся меню **Настройки**, **Панель управления**. На экране появится «Панель управления». Сделать двойной щелчок на пункте **Мультимедиа**;
2) в закладке **Аудио** выбрать в окошках основных устройств воспроизведения и записи любое доступное устройство (куда и откуда Windows по умолчанию будет посылать аудиоданные — звук из воспроизводимого аудиофайла, и с какого устройства будет записывать стандартная программа Windows «Фонограф»);
3) нажать закладку **MIDI**. Выбрать выход MIDI-музыки (куда Windows по умолчанию будет посылать команды, например при воспроизведении MIDI-файла с помощью программы Windows «Универсальный проигрыватель»). Выходом может быть как внутренний синтезатор аудиокарты (например, FM-synthesizer у аудиокарты ESS, или DX-XG-synthe-sizer у звуковой карты Yamaha-724, или Creative Software synthesizer у SBLive или AWE-64), так и внешний (MPU-401 или External MIDI-OUT). В последнем случае MIDI-команды поступают с выхода аудиокарты на вход внешнего синтезатора через MIDI-кабель;
4) по завершении настроек нажать кнопку **OK**.

Для того чтобы открыть системный микшерский пульт (панель, на которой устанавливаются уровни всех источников звука в компьютере), необходимо дважды щелкнуть на значок **Громкоговоритель**

в правом нижнем углу экрана. Появится *микшер* (рис. 4.29) — «смеситель» звуков от различных источников.

Далее следует захватить движок мышью и можно регулировать громкость:

- Volume Control — общая громкость, влияет на все источники звука одновременно;
- Microphone — громкость микрофона, подключенного к входу «Mic» аудиокарты;
- Line — громкость линейного входа (магнитофон, внешний синтезатор);
- CD Audio — громкость CD-ROM при воспроизведении CD Audio; на громкость наушников, подключенных к гнезду на лицевой панели CD-ROM, она не влияет;
- Wave Out — громкость воспроизведения аудиофайлов (.wav, .mp3), т.е. того устройства, которое выбрано как главное в панели **Мультимедиа**, закладка **Аудио**;
- MIDI Out — громкость звучания внутреннего синтезатора аудиокарты, например, при проигрывании MIDI-файлов (.mid), т.е. того устройства, которое выбрано как главное в панели **Мультимедиа**, закладка **MIDI**.

Если на вашем микшере видны не все эти движки, то необходимо:

1) выбрать в меню панели пункт **Параметры**, **Свойства**;
2) в появившейся панели **Свойства** отметить, щелкнув мышкой в квадратике у названия регулятора, в окошке «Отображать регуляторы громкости» нужные вам движки;
3) нажать кнопку **ОК**.

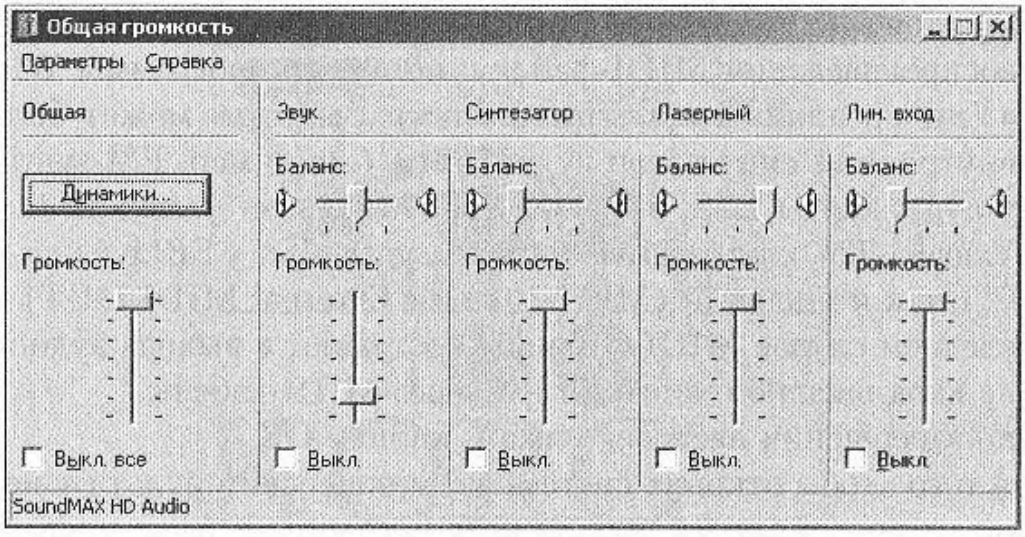
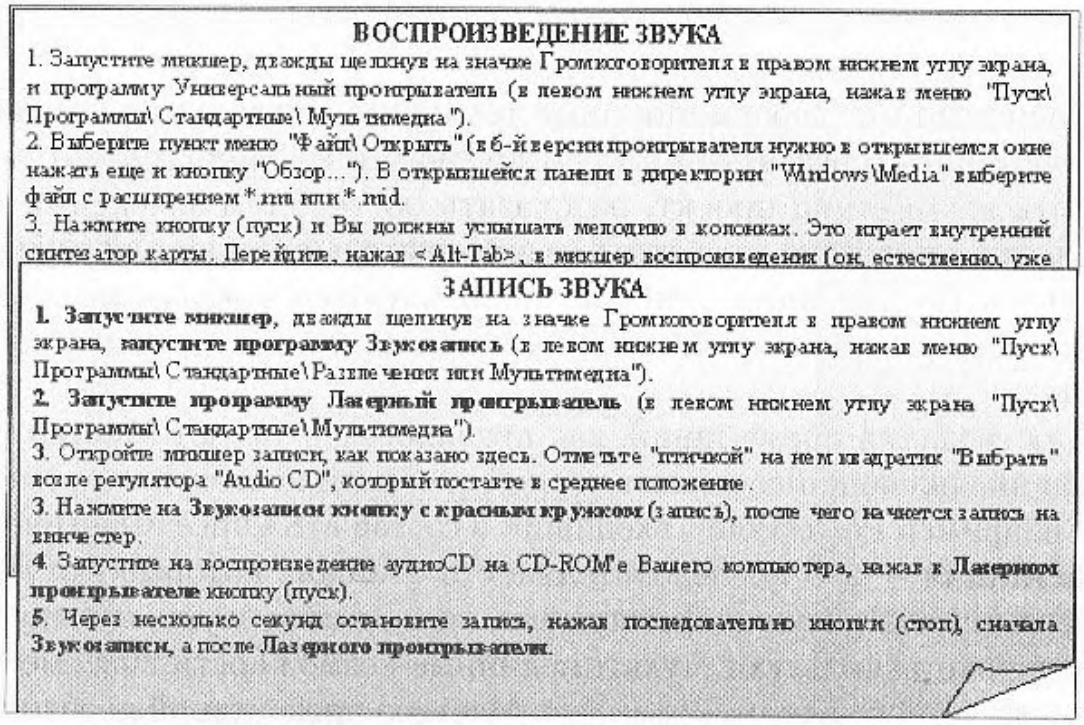


Рис. 4.29. Окно регулировки громкости звука Windows



ВОСПРОИЗВЕДЕНИЕ ЗВУКА

1. Запустите микшер, дважды щелкнув на значке Громкоговорителя в правом нижнем углу экрана, и программу Универсальный проигрыватель (в левом нижнем углу экрана, нажав меню "Пуск\ Программы\ Стандартные\ Мультимедиа").
2. Выберите пункт меню "Файл\ Открыть" (в 6-й версии проигрывателя нужно в открывшемся окне нажать еще и кнопку "Обзор..."). В открывшейся панели в директории "Windows\Media" выберите файл с расширением *.mid или *.mid.
3. Нажмите кнопку (пуск) и Вы должны услышать мелодию в колонках. Это играет внутренний синтезатор карты. Перейдите, нажав <Alt-Tab>, в микшер воспроизведения (он, естественно, уже

ЗАПИСЬ ЗВУКА

1. **Запустите микшер**, дважды щелкнув на значке Громкоговорителя в правом нижнем углу экрана, **запустите программу Звукозапись** (в левом нижнем углу экрана, нажав меню "Пуск\ Программы\ Стандартные\ Развлечения или Мультимедиа").
2. **Запустите программу Лазерный проигрыватель** (в левом нижнем углу экрана "Пуск\ Программы\ Стандартные\ Мультимедиа").
3. Откройте микшер записи, как показано здесь. Отметьте "птичкой" на нем квадратик "Выбрать" возле регулятора "Audio CD", который поставьте в среднее положение.
3. Нажмите на **Звукозаписи кнопку с красным кружком** (запись), после чего начнется запись на винчестер.
4. Запустите на воспроизведение аудиоCD на CD-ROM'е Вашего компьютера, нажав в **Лазерном проигрывателе** кнопку (пуск).
5. Через несколько секунд остановите запись, нажав последовательно кнопки (стоп), сначала **Звукозаписи**, а после **Лазерного проигрывателя**.

Рис. 4.30. Инструкции по воспроизведению и записи звука

Кроме микшера воспроизведения в Windows имеется еще и *микшер записи*. С его помощью можно отрегулировать уровень громкости звука, записываемого на компьютер (например, с помощью программы Windows «Звукозапись»). Для того чтобы его открыть, в той же панели **Свойства** микшера вы можете, щелкнув на кружочке **Запись** и нажав кнопку **OK**, перейти к микшеру записи. Назначение входов аналогично микшеру воспроизведения. Наличие движков на панели записи можно настроить аналогично описанной выше процедуре (при условии, что кружок **Запись** отмечен). Друг на друга эти панели записи и воспроизведения не влияют (рис. 4.30).

? Вопросы и задания

1. Как осуществить запись музыкального произведения с помощью нотного редактора?
2. Какое цифровое оборудование необходимо для обработки звука на компьютере?
3. Как настроить Windows для работы со звуком?
4. Что такое системный микшерский пульт?

П Проектное задание

Запишите короткие интервью с выпускниками и преподавателями. Подготовьте их для размещения на страничке «Ярмарка профессий» (на сайте образовательного учреждения).

4.11. Система компьютерной презентации

Современные информационные технологии позволяют создавать компьютерные презентации. С помощью презентаций можно представить какой-либо проект, рассказать об учебном заведении или конкретном человеке, посвятить ее знаменательной дате или важному событию. Презентации успешно используются в образовательном процессе, при подготовке иллюстраций к докладам, сообщениям и т.д.

Для создания презентаций, как правило, используют **Мастер презентаций**, являющийся компьютерной системой подготовки презентации, например PowerPoint, входящий в состав MS Office. Фактически, **презентация** — это последовательность слайдов, которые могут содержать тексты, рисунки, фотографии, видеоклипы, звуки, музыку, т.е. они широко используют мультимедийные возможности современной компьютерной техники. Для вызова **Мастера презентаций** необходимо выполнить команды **Пуск**, **Программы**, **Microsoft Office** и щелкнуть по строке **Microsoft Power Point**, — появится окно (рис. 4.31).

Из меню в правой части сначала требуется выбрать макет слайда, который будет определять расположение в слайде текста и графики.

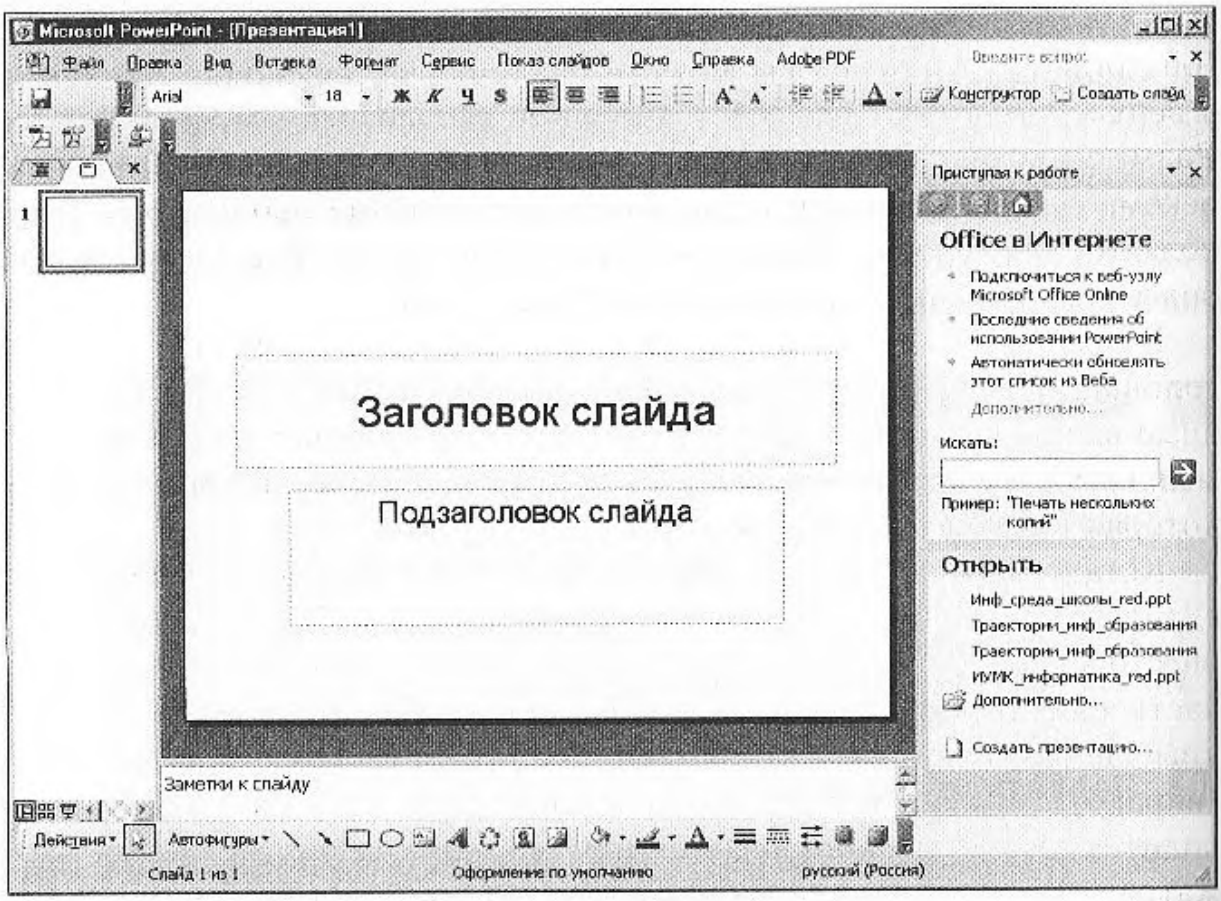


Рис. 4.31. Основное окно MS Office PowerPoint

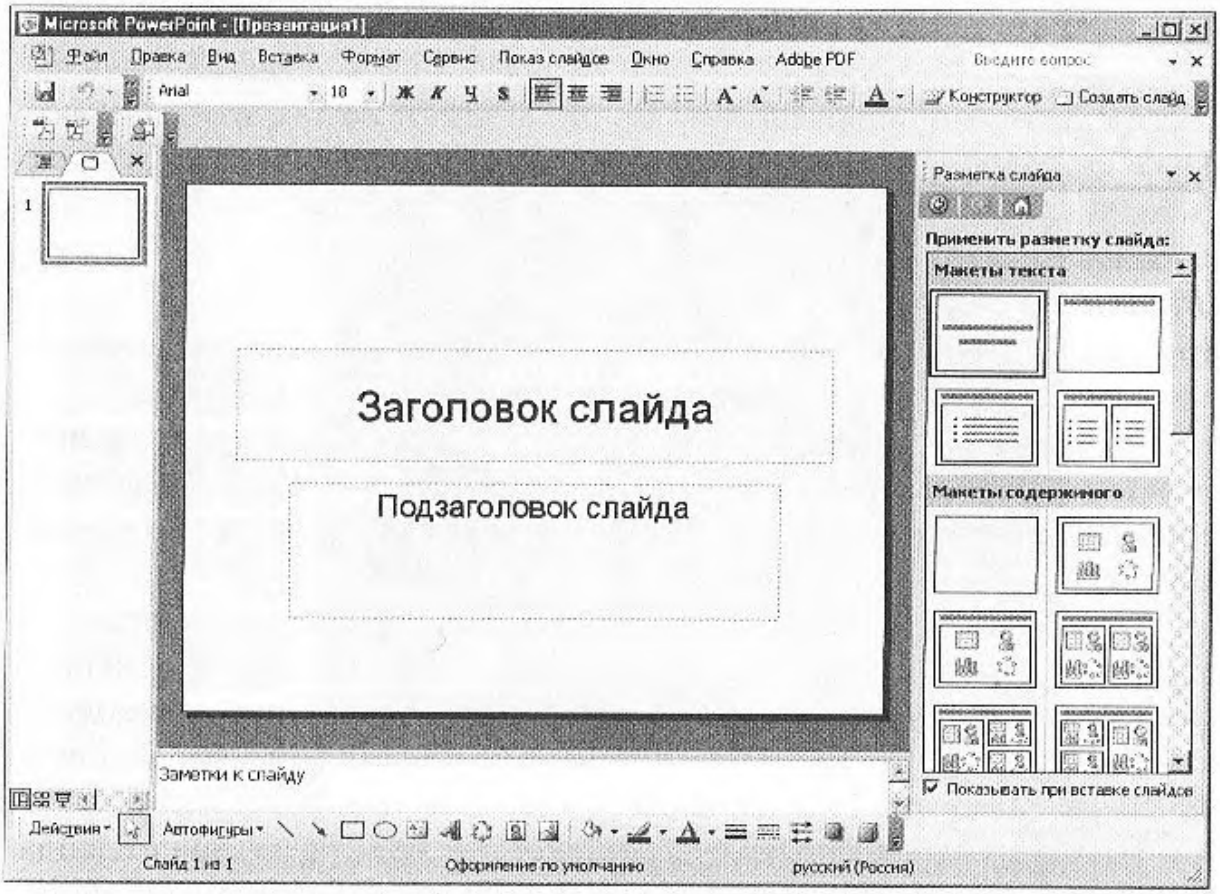


Рис. 4.32. Разметка слайда

Для того чтобы это сделать, надо нажать на кнопку **Конструктор**, затем в верхней части появившейся панели в выпадающем меню выбрать **Разметка слайда** (рис. 4.32). Следует сказать, что эти макеты носят рекомендательный характер, и, в принципе, можно произвольно располагать информацию на слайде, однако чтобы презентация получилась более эффектной, желательно их придерживаться.

В правой части вы увидите самые разнообразные макеты. Для заголовка презентации нужно выбрать один из них, а затем шаблон дизайна презентации (для этого в верхней части панели в выпадающем меню выбрать **Дизайн слайда**), который определит ту цветовую гамму, которая будет использоваться (рис. 4.33).

Кстати, выбранный дизайн можно применять как к отдельным слайдам, так и ко всем входящим в презентацию. Далее необходимо внести заголовок и подзаголовок нашей презентации. Затем в верхней части панели в выпадающем меню выбрать **Смена слайдов** и в открывшемся окне указать, как будет меняться слайд (уходить влево (вправо), закрываться жалюзи и т. п.), а также сколько времени отводится на его экспонирование, или же переход определить по нажатию любой клавиши (это особенно удобно при проведении докладов) (рис. 4.34).

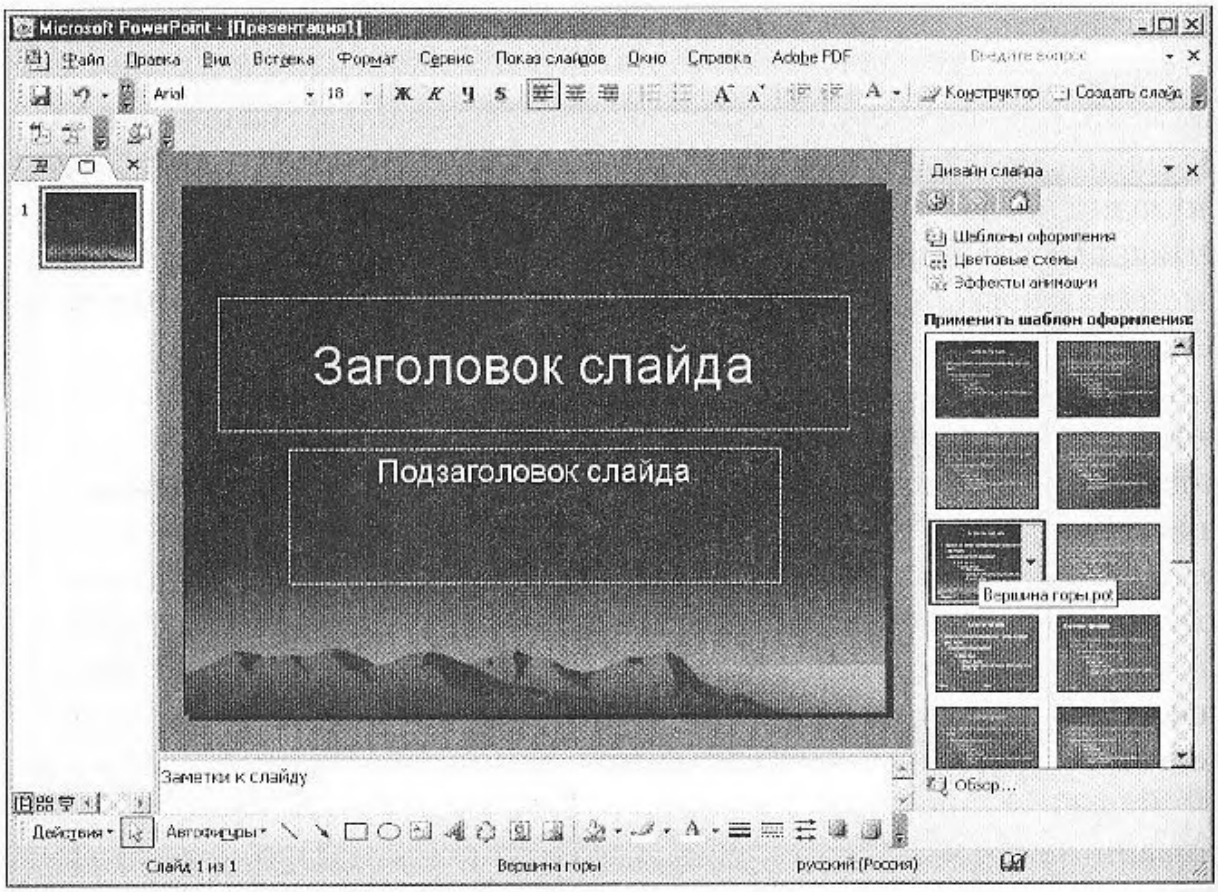


Рис. 4.33. Дизайн слайда

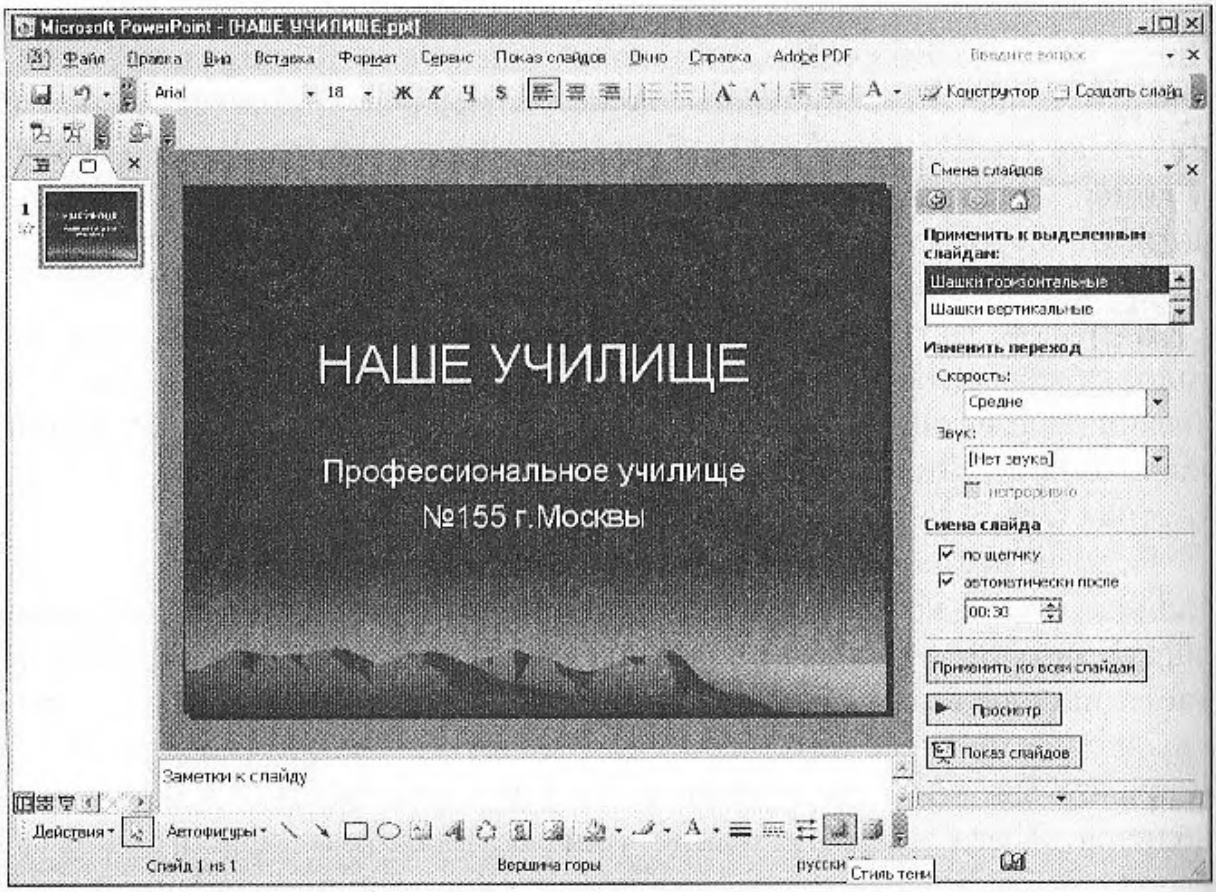


Рис. 4.34. Смена слайдов

Установим, что смена слайда происходит через «Шашки горизонтальные», со средней скоростью, и слайд экспонируется в течение 30 с. Отметим, что в любой слайд, используя команду **Вставка**, можно вставить рисунок, звук, видеосюжет из соответствующего файла. Можно записать дикторский текст, используя возможность «Звукозапись». Итак, первый слайд готов.

Для создания следующего слайда необходимо выполнить **Вставка**, **Создать слайд**. В появившемся макете (а вы можете выбрать любой) написать вверху «Наше училище образовано в 1965 г.». Затем выполнить **Вставка**, **Рисунок**, **Из файла** и указать на файл, где уже приготовлена заранее фотография здания нашего училища. При этом указать, что смена слайда будет происходить с помощью эффекта «Прямоугольники внутрь», который мы выбрали, и установить среднюю скорость смены слайда и время экспонирования 15 с (рис. 4.35).

Если у вас есть звуковой файл с какой-нибудь подходящей по случаю музыкой, можно вставить и его, тогда эта музыка будет звучать все время экспонирования слайда.

Далее следует опять выполнить **Вставка**, **Создать слайд**, сменить дизайн, сделать заголовок «Наш директор», вставить фотографию директора и набрать сопроводительный текст (рис. 4.36).

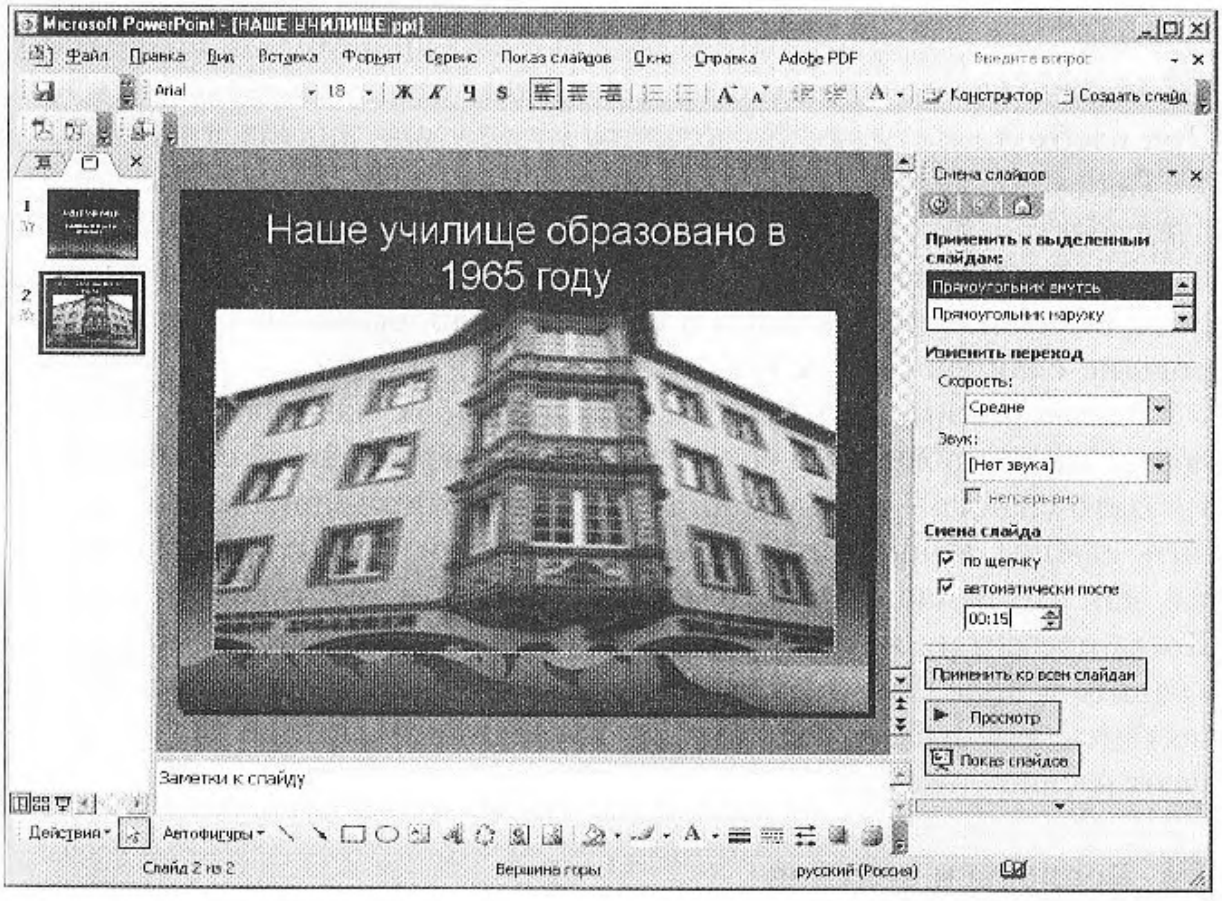


Рис. 4.35. Слайд 2

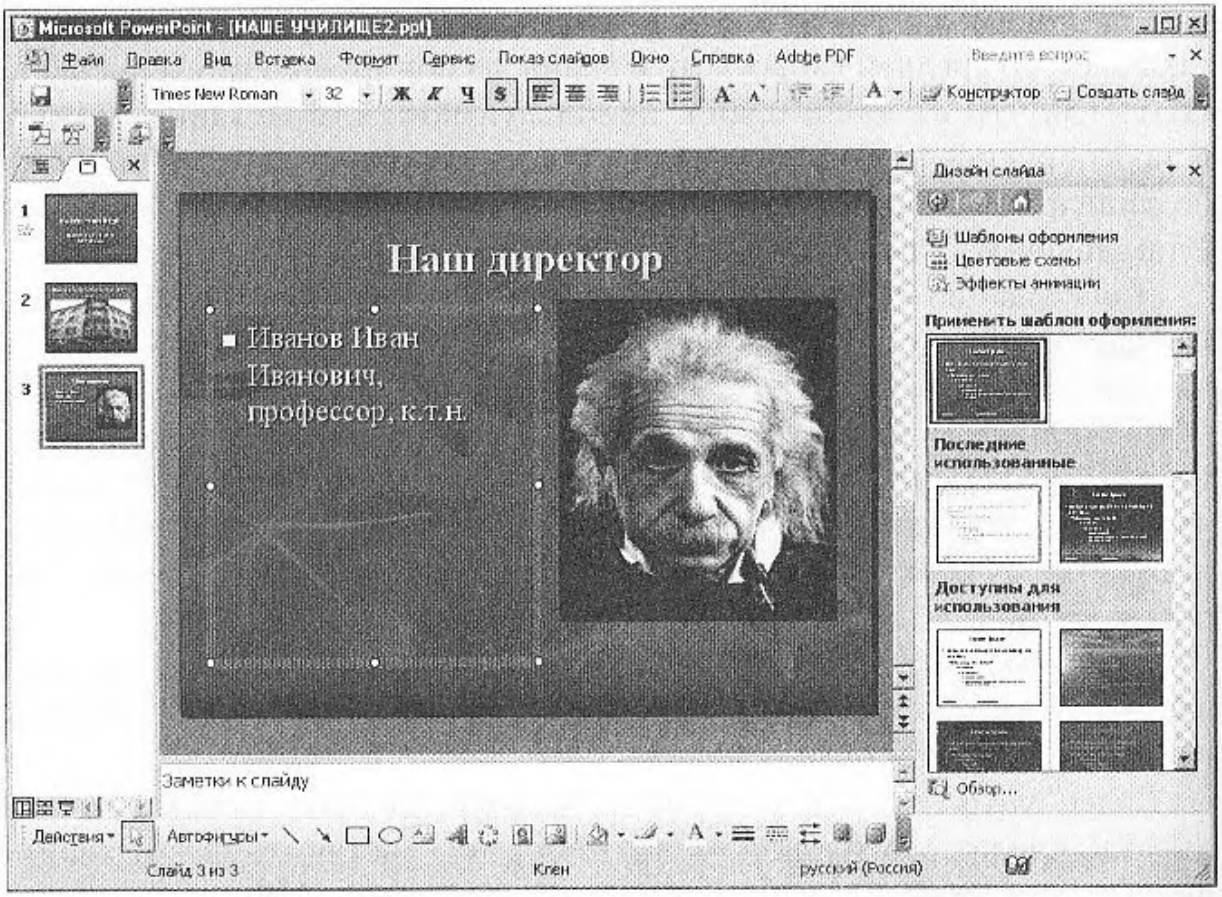


Рис. 4.36. Слайд 3

Определим, что смена этого слайда будет происходить по эффекту «Наплыв вниз», со средней скоростью, и он будет экспонироваться 15 с. При необходимости можно вставить музыку или дикторский текст.

Таким образом, в нашей презентации уже три слайда. Так же следует действовать и далее. Для просмотра того, что получилось, нужно выполнить **Показ слайдов**, **Начать показ**. Для того чтобы удобнее смотреть состав презентации в целом, можно выполнить **Вид**, **Сортировщик слайдов** (рис. 4.37).

В этом режиме под каждым слайдом слева указывается заданное время экспонирования, а справа его номер; вертикальная линия правее третьего слайда — это курсор. В режиме сортировщика можно добавлять слайды, копировать их, менять местами, удалять примерно так же, как это делается с фрагментами текста в текстовом редакторе. Когда презентация готова (или в процессе подготовки), ее можно сохранить в виде файла на диске, задав соответствующее имя. Нужно помнить, что файлы, содержащие презентации, автоматически получают расширение .ppt.

Вопросы и задания

1. Что представляет собой компьютерная презентация?

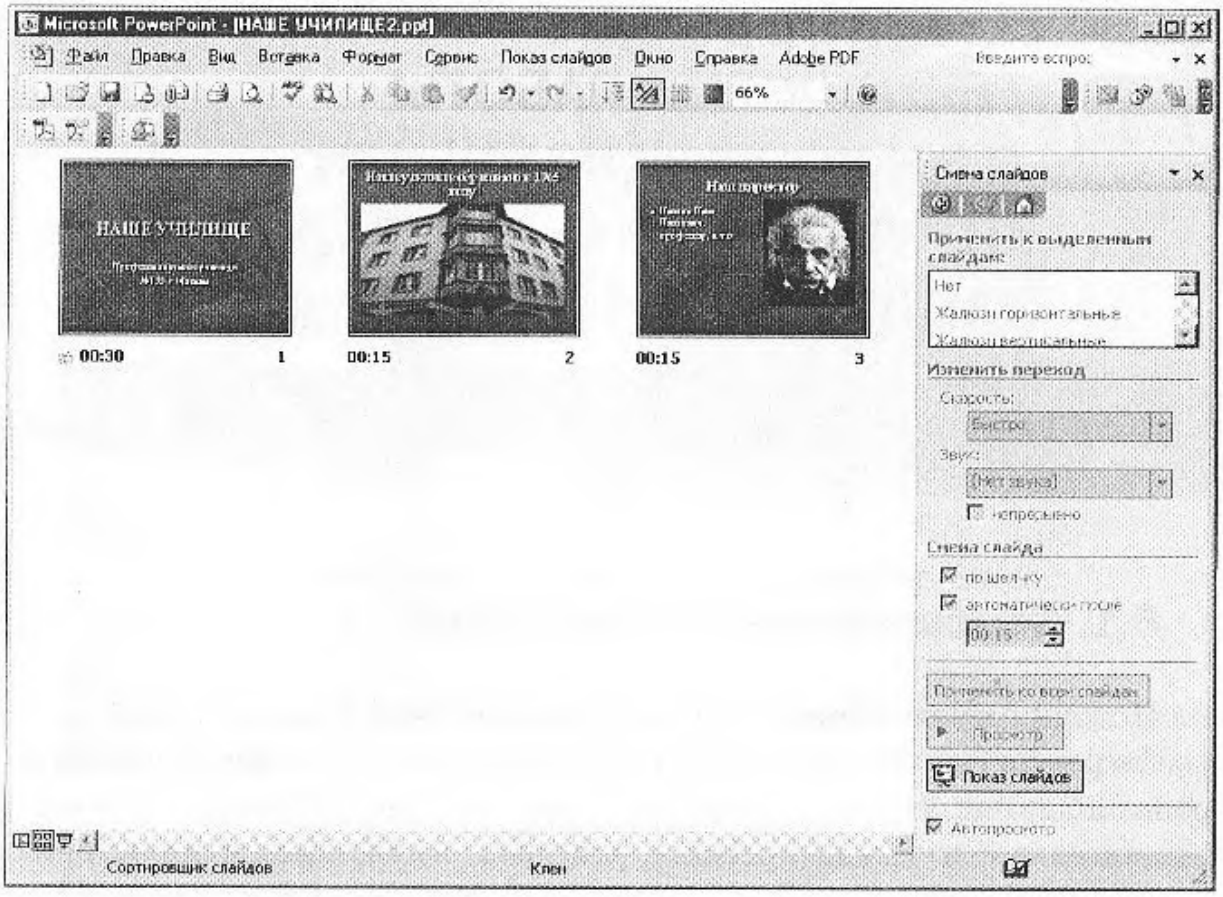


Рис. 4.37. Сортировщик слайдов

2. Почему можно рассматривать компьютерную презентацию как важное коммуникативное средство?
3. Опишите технологию создания компьютерной презентации.
4. Используя описанную технологию создания презентации, выполните задание «Визитная карточка» для вашего образовательного учреждения. Используйте текстовые, графические, звуковые, фото- и видеоматериалы по теме.

Проектное задание

Выполните презентацию к докладу по защите своего курсового проекта.

ГЛАВА 5 ТЕХНОЛОГИИ РАБОТЫ С ИНФОРМАЦИОННЫМИ СТРУКТУРАМИ — ЭЛЕКТРОННЫМИ ТАБЛИЦАМИ И БАЗАМИ ДАННЫХ

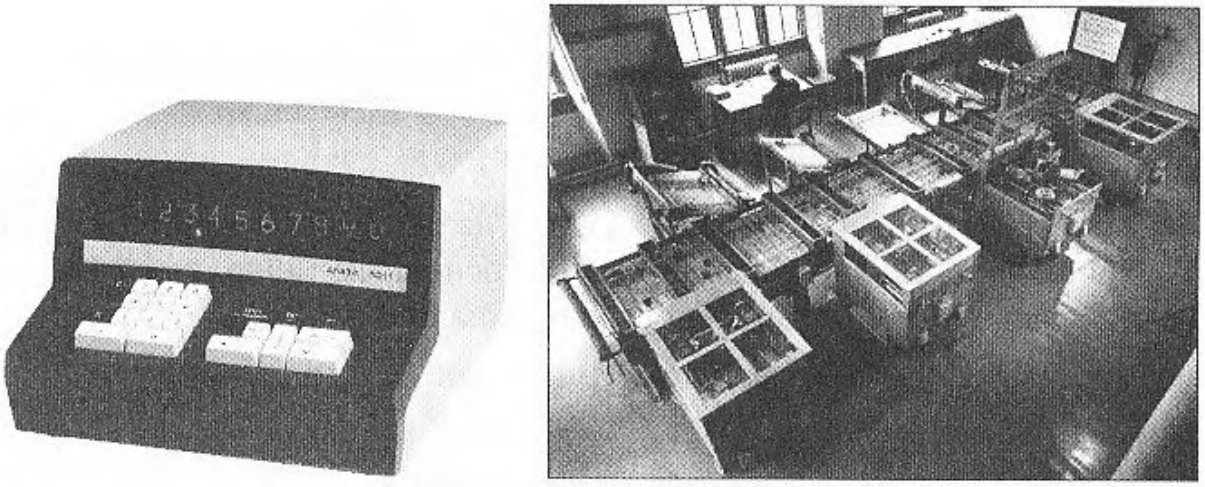
5.1. Компьютер как вычислитель

История вычислений — это история развития знаний. Наверное, необходимость счета возникла у человека как основа общения. Природа подарила человеку своеобразные счетные палочки — пальцы. Удивительно, но каждый ребенок повторяет в своем развитии историю вычислений, от пальцев — к современным супервычислителям. Проследим историческую цепь создания вычислительных устройств: подручные материалы (пальцы, палочки, камешки и пр.), счеты, абаки, арифмометры, калькуляторы, компьютеры (рис. 5.1).

> Какие арифметические операции умеет выполнять компьютер и где они выполняются? Назовите порядок операций для расчета алгебраического выражения, докажите на примере, что нарушение очередности операций ведет к ошибке. Используя алгебраическое выражение, проставьте в нем порядок операций и проведите вычисления на калькуляторе. Что вы можете сказать о точности вычислений? Что такое абсолютная погрешность и как она определяется?

Все команды компьютера выполняются в процессоре, обрабатываемые данные при этом находятся в оперативной памяти. Чтобы рассчитать формулу, необходимо точно соблюдать порядок операций — алгоритм расчета. Действия — вычисления в алгоритме — выполняются в строгом порядке, последовательно друг за другом. Такой алгоритм расчета называется *линейным*.

В каждом действии алгоритма вычисления формулы участвуют два операнда, над которыми производятся действия сложения, вычитания, умножения или деления. Кроме того, к данным может быть применена функция, например, возведения в степень или преобразования в дробь,



а *б*

Рис. 5.1. Примеры первых электронно-вычислительных устройств:
а — калькулятор Anita; *б* — Кембриджский дифференциальный анализатор

обратную числу. Результат любого вычисления может использоваться для другого расчета. Вы, конечно, знакомы с простейшим электронным вычислителем — калькулятором. Почему он так называется? *Калькулятор* (от лат. *calculator* — счетчик) умеет только считать и сохранять результат в своей небольшой оперативной памяти. При выключении калькулятора его память «сбрасывается» в ноль.

В случае использования компьютера не нужно иметь калькулятор, так как в каждом компьютере есть программа, реализующая его работу. Более того, даже меню этой программы сделано очень похожим внешне на калькулятор, так что познакомиться с ним будет совсем несложно (рис. 5.2). Можно сказать, что программа-калькулятор — это простейший числовой редактор, который позволяет ввести число, исправить его, выполнить вычисление с ним и вывести результат на экран дисплея калькулятора.

Используя пиктограмму **Пуск**, выберите команды **Программа**, **Стандартные**, **Калькулятор**.

Набор числа осуществляется с помощью мыши и кнопок-цифр. Есть на панели такого калькулятора кнопки с обозначением операций «минус», «плюс», «умножение», «деление» и некоторых функций. В программе-калькуляторе, как и в самом устройстве, присутствуют специальные кнопки управления памятью, т. е. ячейками, в которых можно сохранить результат расчета:



Рис. 5.2. Калькулятор Windows

- MR— просмотр на экране содержимого памяти;
- MS — сохранение в памяти промежуточного результата расчета;
- MC — очистка памяти.

Памятью удобно пользоваться для сохранения промежуточных результатов, которые используются для дальнейших вычислений.

Кроме того, можно воспользоваться основным меню, которое, как всегда, расположено в верхней строке окна (табл. 5.1).

Таблица 5.1

Команда меню	Описание
Правка	Позволяет скопировать фрагмент числа, формулы в промежуточную память и восстановить из нее на однострочном экране калькулятора
Вид	Дает возможность выбрать один из двух калькуляторов, заложенных в данную программу: обычный и инженерный

Очистите память и экран калькулятора и выполните действия по алгоритму вычисления предложенной функции в режиме обычного калькулятора для значений x и y:

$$y = \frac{1}{(x+1)(x+2)} - 5\frac{x}{x-5} + 4x^2.$$

Вариант	1	2	3
x	0,3	–5,4	1,1
y	1,01	114	6,44

Алгоритм расчета на калькуляторе представлен следующими шагами, где последовательно перечислены кнопки для выполнения команды (знак ⇒ обозначает запись в память):

1. $x + 1 = \Rightarrow$ MS
2. $x + 2 =$
3. * MS =
4. Применить к результату функцию $1/x$ и сохранить в другой ячейке памяти, если она есть на калькуляторе, или записать в блокнот.
5. $x - 5 \Rightarrow$ MS
6. $5 * x =$
7. / MS ⇒ MS
8. $x * x =$

9. * 4 =
10. –MS =
11. + ответ шага 4 =

Сравните свой ответ со значением y из таблицы. Добейтесь правильного ответа. Внимание, знак «=» означает выполнение команды на калькуляторе, результат которой мы увидели на экране.

Вы научились вычислять формулы на калькуляторе, используя правила вычислений. Но компьютерный вычислитель позволяет рассчитать всю функцию сразу, для этого ее нужно записать по правилам компьютерного вычислителя. Примерами *компьютерного вычислителя* являются калькулятор формул, электронная таблица и др.

Инженерный калькулятор помимо библиотеки функций, необходимых в технических расчетах, позволяет не только вводить числа для расчета и указывать операции над ними, как в обычном калькуляторе, но и записывать формулы в промежуточную память. Набор осуществляется кнопками калькулятора последовательно друг за другом с указанием чисел, функций и скобок.

Воспользуемся инженерным калькулятором для расчета алгебраических выражений, которые после записи по правилам компьютерного вычислителя называются *арифметическими выражениями*.

Вот эти правила:

1) знаки арифметических операций оформляются символами «+» (сложение), «–» (вычитание), «*» (умножение), «/» (деление), «^» (возведение в степень);
2) установка очередности операций оформляется с помощью круглых скобок. Количество открывающих и закрывающих скобок должно быть одинаковым;
3) математические функции записываются с помощью специального имени, аргумент заключается в скобки, например:

- SQR(x) — корень квадратный;
- ABS(x) — абсолютная величина;
- INT(x) — целая часть от числа.

Арифметическое выражение описывает линейный алгоритм расчета формулы. Запишем алгебраическое выражение в виде арифметического — линейного алгоритма расчета. Будем использовать скобки в целях определения очередности операций. Для $x = \pi$

$$y = \frac{1}{(x+1)(x+2)} - 5\frac{x}{x-5} + 4x^2.$$

В данном алгебраическом выражении использована дробная запись. В компьютерном вычислителе она не предусмотрена, но мы можем представить дробь как операцию деления числителя на знаменатель. Чтобы не ошибиться в порядке операций, будем придерживаться правила записи формулы, представленной дробью: заключенный в скобки числитель делим на заключенный в скобки знаменатель.

Получится следующее арифметическое выражение (обратите внимание, как скобки управляют порядком вычислений в этом линейном алгоритме расчета):

$y = (1)/((x + 1)*(x + 2)) - (5*x)/(x - 5) + 4*x\hat{}2.$

Порядок операций

4 1 3 2 8 5 7 6 11 10 9

Стрелка показывает направление вычислений, а числа — порядок операций. Помните, что функции всегда выполняются раньше арифметических операций.

? Вопросы и задания

1. Что представляет собой программа-калькулятор? Перечислите основные элементы калькулятора. Объясните их назначение.
2. Что такое арифметическое выражение и каковы правила его оформления? Выполните последний пример, используя калькулятор мобильного телефона.
3. В каком устройстве компьютера производится расчет арифметического выражения?
4. Как мы управляем порядком операций в записи линейного алгоритма вычислений — арифметическом выражении?

П-СЭ Проектное задание

Подготовьте доклады по темам, указанным в табл. 5.2, используя Интернет-энциклопелию www.megabook.ru и поиск по темам.

Таблица 5.2

Тема	План доклада
Абак — ручной вычислитель	Счетная скамья (банк) и чекерд (клетчатый), от которых произошли слова банк и чек — абаки. Абак у египтян, греков и римлян. Квипос (узелки на память) у обитателей Южной Америки. Суанпан — китайские семикосточковые счеты, соробан — японские семикосточковые счеты. Счеты в России

Окончание табл. 5.2

Тема	План доклада
Создатели арифмометра	Блез Паскаль и его вычислительная механическая машина — паскалина (1642 г.). Готфрид Вильгельм Лейбниц — автор арифмометра (1673 г.)
Первый компьютерщик — Чарльз Бэббидж и первая программистка — Ада Байрон (Лавлейс)	Вычислительная машина Бэббиджа (1823 г.). Программа вычислений и память для результатов. Процессор — «сердце» машины, которая выполняет программу, использующую результаты из памяти. Язык программирования — язык, понятный машине. Язык описания команд для машины Бэббиджа, предложенный Адой Байрон
Калькулятор и компьютер	Калькулятор как устройство с временной оперативной памятью, осуществляющее только счет по одной, указанной человеком в данный момент команде. Вычислительная среда компьютера — программная среда, выполняющая программу, написанную человеком, состоящую из множества команд для вычислений, и предназначенная для многократного использования с различными исходными данными. Можно ли сравнивать числа на калькуляторе? Компьютер — устройство, имеющее процессор для арифметических и логических операций над числами

5.2. Моделирование электронной таблицы

Вычислительная среда, позволяющая строить таблицы для статистической обработки данных, т.е. включающая в себя библиотеку математических, коммерческих и логических функций, а также графические возможности демонстрации статистических процессов во времени, называют **электронной таблицей**. В ней можно формировать большие массивы числовых данных для их дальнейшей обработки, упорядочивать эти данные по возрастанию и убыванию, находить суммарные и средние арифметические значения, строить графики и диаграммы, пересчитывать данные по указанной формуле, делать выборку данных, которые отвечают некоторым условиям.

Такие таблицы издавна используются человеком, но они формировались в бумажном виде, а расчет производился с помощью счетов, арифмометров и калькуляторов. В настоящее время в статистических

расчетах применяется компьютерная вычислительная среда, способная выполнить работу быстро и без ошибок, а также настроить данные на новые значения ключевых параметров, например курс валюты на конкретный момент времени или средний проходной балл на конкретный день вступительных экзаменов и пр.

Бухгалтерия, банки, налоговые инспекции, кассы в магазинах, статистические БД учебных учреждений, страховых компаний, магазинов, оптовых баз, строительных компаний, складов, авиационных и железнодорожных касс используют возможности компьютера для проведения объемных, быстро меняющихся, трудоемких по обработке операций с числовой информацией.

Главными понятиями при работе с электронными таблицами являются: ячейка, столбец и строка, адрес, тип данных в ячейке, диапазон, выделение диапазона, редактирование данных в таблице, расчет данных диапазона, построение графиков и диаграмм для диапазона.

Для того чтобы познакомиться с правилами формирования таблицы данных, достаточно вспомнить игру «Морской бой». Каждой клетке (ячейке) поля игры соответствует адрес — буква и цифра. Так и в электронной таблице. Каждая *ячейка* имеет *адрес*: первой указывается буква — пометка столбца, а затем цифра — пометка строки от левого верхнего края таблицы. Ячейка находится на пересечении столбца и строки, поэтому ее адрес состоит из двух пометок. Например, на рис. 5.3 ячейка, в которой установлено число 35, имеет адрес B2.

Столбец помечается адресом-буквой, а *строка* — адресом-цифрой. Набор нескольких ячеек одной строки задается *диапазоном*. Например, A1:A10 представляет собой диапазон от 1 до 10 клетки включительно, находящихся в столбце A.

Адрес ячеек, записанный с помощью буквы и цифры, называется *относительным адресом*, так как он постоянно соотносится со столбцом и строкой вслед за перемещением курсора. Адрес «бежит» вслед за курсором, высвечивая столбец и строку. При копировании данных из строки в строку, из столбца в столбец, из клетки в клетку их адреса будут переустанавливаться автоматически, т. е. соотносится с конкретной строкой и столбцом. Переписывая новые адреса, электронная таблица копирует данные в них без изменения.

	A	B
1		
2		35
3		
4		

Рис. 5.3. Фрагмент электронной таблицы

Данные в ячейках могут иметь разный тип, быть текстом, числом и формулой. Для облегчения нашего труда электронная таблица самостоятельно оформит данные как текст, если они начинаются с буквы. *Формула* для расчета набирается после знака равенства. Его указание обязательно! А вот числа набираются сразу, как

мы делали это на калькуляторе, используя знак «+» и «–», а также десятичную точку.

Данные можно располагать в ячейках прижатыми вправо, влево и по центру, а числам можно определить длину десятичной части. Кроме того, числа можно представлять в денежном и процентном виде.

Основные команды меню электронной таблицы приведены в табл. 5.3, а вид экрана на примере приложения Microsoft Excel — на рис. 5.4.

Таблица 5.3

Меню	Команда	Описание
Файл	Создать	Создает новую таблицу
	Открыть	Открывает существующую таблицу
	Закрыть	Закрывает файл с таблицей
	Сохранить	Сохраняет таблицу на диске в файле с тем же именем, что и ранее
	Сохранить как...	Сохраняет таблицу на диске в файле с заданным именем
	Параметры страницы	Позволяет задавать размер полей, а также ориентацию страницы — книжную или альбомную
	Предварительный просмотр	Позволяет предварительно посмотреть вид страницы при ее печати
	Печать	Позволяет напечатать страницу или выделенный диапазон ячеек
Правка	Вырезать	Вырезает выделенный фрагмент и помещает его в буфер
	Копировать	Копирует выделенный фрагмент в буфер

Продолжение табл. 5.3

Меню	Команда	Описание
Правка	Вставить	Вставляет содержимое буфера
	Специальная вставка	Вставляет формулы, значения или форматы фрагмента из буфера
	Найти	Ищет заданный объект
	Заменить	Ищет заданный объект и заменяет его
Вид	Панели инструментов	Открывает панели инструментов: стандартную, редактирования, рисования и др.
	Строка редактирования	Открывает (закрывает) строку редактирования
	Строка состояния	Открывает (закрывает) строку состояния
	Масштаб	Изменяет масштаб представления данных
Вставка	Ячейки	Позволяет вставить ячейку (группу ячеек)
	Строки	Позволяет вставить строку (группу строк)
	Столбцы	Позволяет вставить столбец (группу столбцов)
	Диаграмма	Позволяет вставить диаграмму
	Функция	Позволяет вставить функцию из набора имеющихся
	Имя	Позволяет присвоить имя конкретной ячейке

Продолжение табл. 5.3

Меню	Команда	Описание
Вставка	Примечание	Позволяет прокомментировать назначение или содержание ячейки (в правом верхнем углу — красная точка)
	Рисунок	Позволяет вставить рисунок из имеющейся коллекции или из произвольного файла
	Объект	Позволяет вставлять различные объекты: редактор формул, видео, звук, текст и т.д.
Формат	Ячейки	Устанавливает формат ячейки
	Строка	Устанавливает формат строки (строк)
	Столбец	Устанавливает формат столбца (столбцов)
	Лист	Устанавливает формат листа
Сервис	Орфография	Позволяет проверять орфографию текста
	Защита	Позволяет защищать лист паролем
	Макрос	Позволяет вводить макросы (макрокоманды)
	Параметры	Позволяет устанавливать параметры таблицы
Данные	Сортировка	Позволяет сортировать данные
	Внешние данные или импорт внешних данных	Позволяет подключать внешние данные, в частности из БД

Меню	Команда	Описание
Окно	Новое	Открывает новое окно
	Расположить	Устанавливает порядок расположения нескольких окон (рядом, каскадом и т. д.)
	Скрыть	Скрывает окно
	Разделить	Обеспечивает вертикальное или горизонтальное разделение окна

Прежде чем начать работу с таблицей, нужно хорошо подготовиться к этому, в частности определить размеры таблицы, размеры строк и столбцов, форматы представления чисел, шрифты, используемые цвета, порядок выравнивания данных, расположение таблицы (на одном или нескольких листах), оформление границ и т. д. Именно с этого процесса, который называют *моделированием электронной таблицы*, собственно, и начинается работа.

Размеры ячеек, строк, столбцов определяют с помощью меню **Формат** (рис. 5.5) или соответствующих команд контекстного меню и пиктограмм.

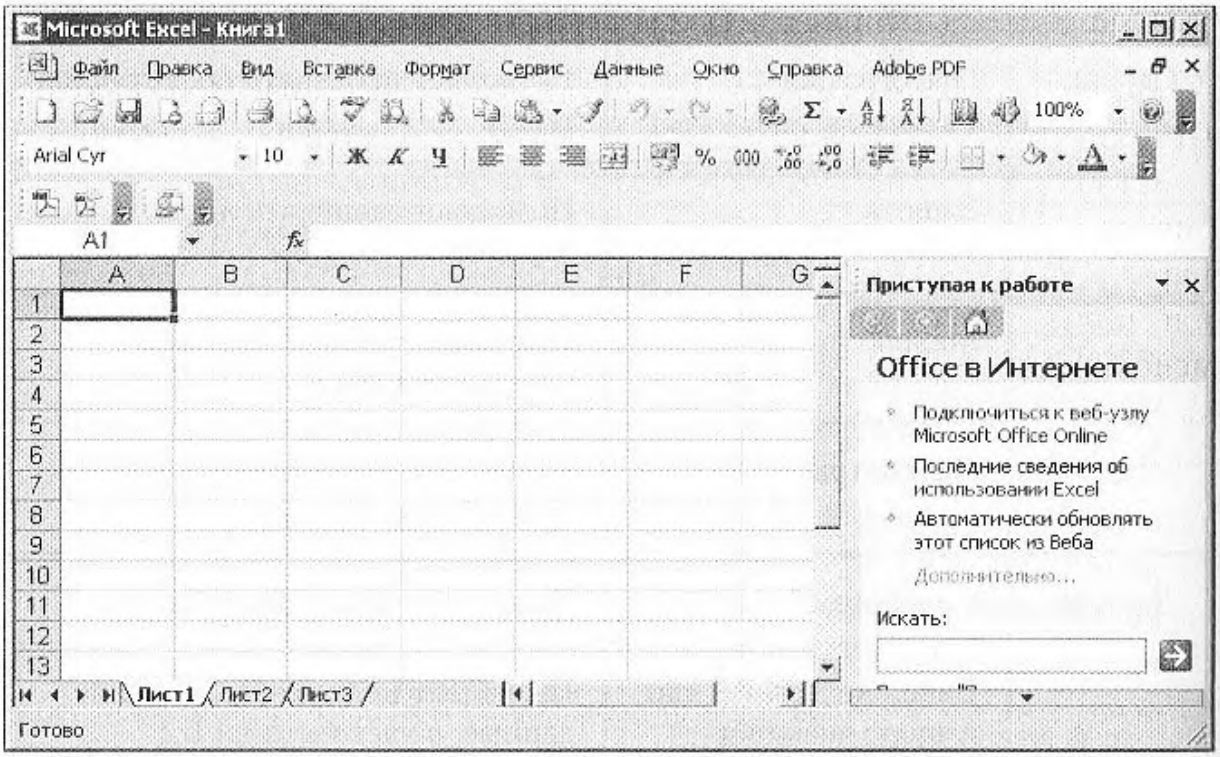


Рис. 5.4. Электронная таблица MS Excel Windows

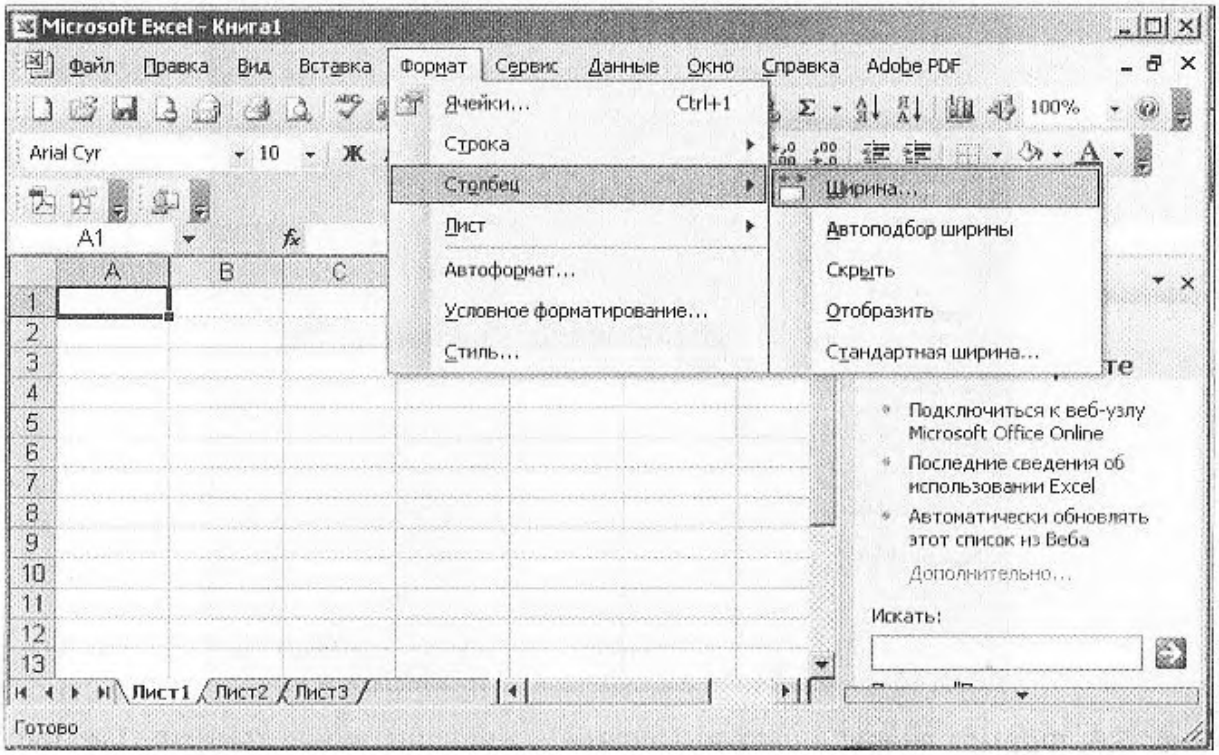


Рис. 5.5. Меню «Формат»

Изменить размер можно одним из следующих способов:

- буксировка за границу раздела номеров строки или букв столбца;
- автоподгон размера — двойной щелчок на границе раздела;
- команда **Строка (Столбец)**, далее **Высота (Ширина)** или **Автоподбор**;
- **Стандартная Ширина** в подменю **Столбец**.

Высота строки измеряется в пунктах, так же как и размер символов. Ширина столбца измеряется количеством стандартных символов.

Строки и столбцы можно скрыть для удобства визуального просмотра или печати или показать. Перед отображением требуется замаркировать соседние со скрытыми строки или столбцы. Введенное или полученное в результате расчета число можно представить в измененном виде. Это выполняется по команде **Формат**, **Ячейки**, вкладка **Число** (рис. 5.6).

Выделяют следующие виды форматов:

- *общий* — представление числа в том виде, в котором оно было введено в таблицу;
- *числовой* — представление числа с дробной частью или в виде целого, а также отрицательное — красным цветом;
- *денежный и финансовый* — формы числа с выводом денежной единицы в виде целого или с дробными знаками после запятой, а также отрицательное число красным цветом;

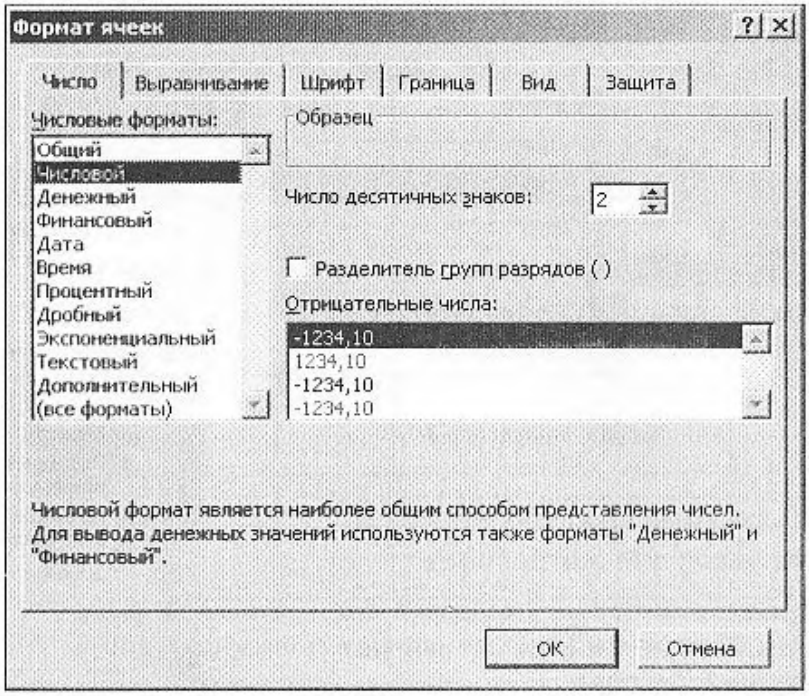


Рис. 5.6. Формат ячеек: вкладка «Число»

- *экспоненциальный* — для больших или очень маленьких чисел, например в виде 25E–4;
- *процентный и дробный* — представление числа в процентах (*100) и в виде десятичной дроби;
- *дата и время* — варианты представления;
- *текстовый* — число представляется в виде текстовой константы;
- *дополнительный* — почтовый индекс, номер телефона, табельный номер.

На панели форматирования имеются кнопки для отображения числа в денежном представлении, в виде процента, с разделителями тысяч и для увеличения и уменьшения значащих цифр после запятой.

По горизонтали информацию можно расположить (рис. 5.7):

- обычно (по умолчанию текст слева, число справа);
- по левому или правому краю ячейки;
- по центру ячейки;
- с заполнением всей области ячейки путем повторения содержимого;
- по обоим краям;
- по центру выделенной области (по выделению).

Вертикально информацию можно разместить:

- по верхнему краю;
- по центру;
- по нижнему краю;
- по обоим краям.

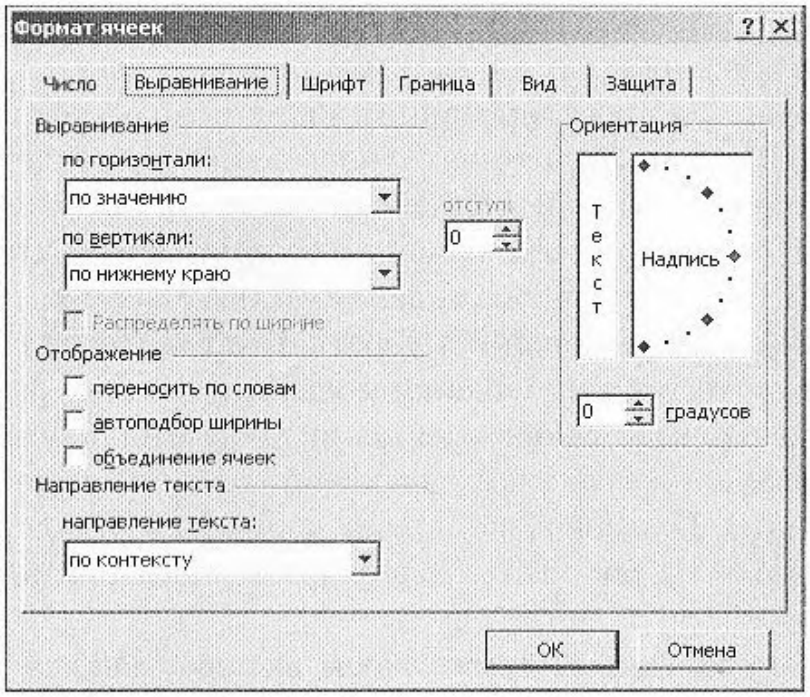


Рис. 5.7. Формат ячеек: вкладка «Выравнивание»

Ориентировать содержимое ячейки можно по вертикали и под углом. Для любой ориентации длинный текст ячейки можно *переносить по словам* на новую строку, образуя при этом одну ячейку.

Для изменения шрифта можно воспользоваться командой **Формат ячеек**, вкладка **Шрифт** либо применить соответствующие кнопки на панели **Форматирование** (рис. 5.8). Используя эту команду, можно

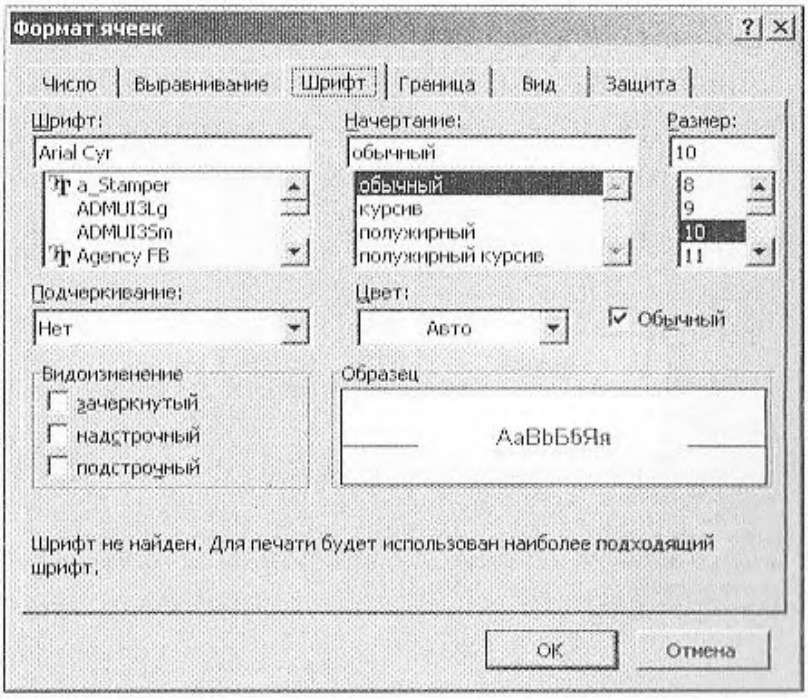


Рис. 5.8. Формат ячеек: вкладка «Шрифт»

также подчеркнуть текст, назначить эффекты (зачеркнутый, верхний индекс — степень, нижний индекс) и установить цвет.

Очень часто при проектировании таблиц используют вкладку **Граница** (рис. 5.9). Данный элемент оформления позволяет обрамлять табличные данные внутри и снаружи (вокруг) выбранным стилем линии сетки. Перед вызовом команды **Формат ячеек**, **Граница** необходимо выделить область обрамления. Для удаления линии рамки нужно выделить левую ячейку и убрать отметку рамки справа, затем выделить правую соседнюю ячейку и удалить отметку рамки слева.

В команде **Формат ячеек** имеется вкладка **Вид**, с помощью которой можно выполнить заливку выделенной области ячеек выбранным цветом и узором. Пример выполнения заливки столбцов различными цветами показан на рис. 5.10. Содержимое ячеек при этом не теряется.

С помощью команды **Формат ячеек**, вкладка **Защита** можно установить режим защиты ячеек, скрыть формулы (рис. 5.11). По умолчанию все ячейки листа защищены, поэтому для снятия блокировки с некоторой зоны ее нужно выделить и убрать флажок на вкладке **Защита**. *Защищенные ячейки* не могут быть отредактированы, удалены и перенесены. *Скрыть формулы* — означает отмену их отображения в строке формул.

Указанные меры защиты начинают действовать только после команды **Сервис**, **Защита**, **Защитить лист**. В ней пользователь может ввести пароль защиты, который будет запрашиваться при снятии защиты по той же команде **Защита**.

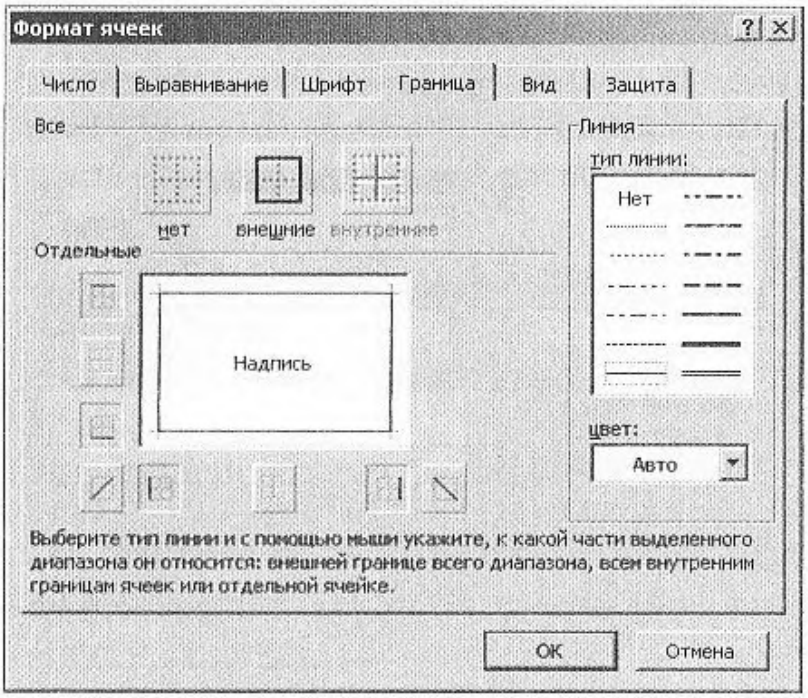


Рис. 5.9. Формат ячеек: вкладка «Граница»

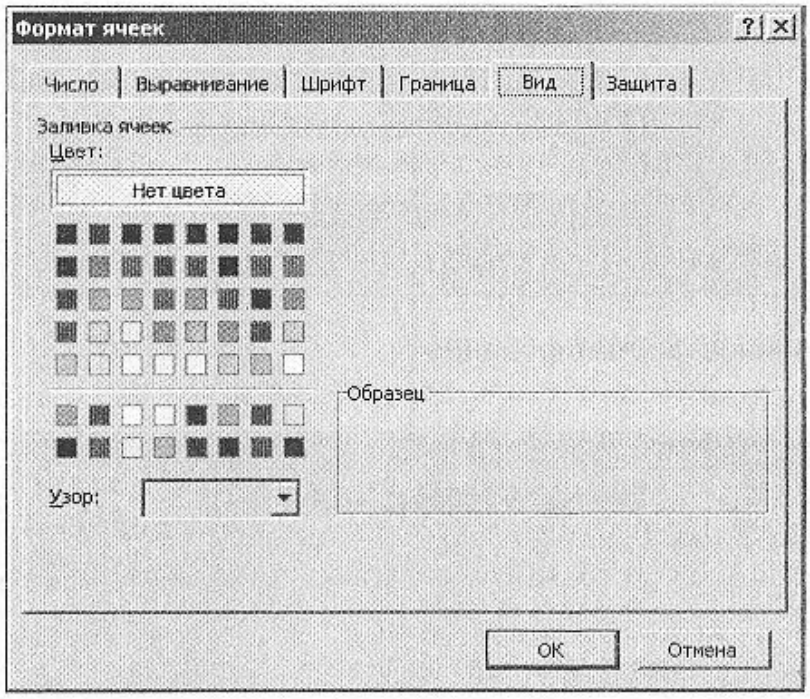


Рис. 5.10. Формат ячеек: вкладка «Вид»

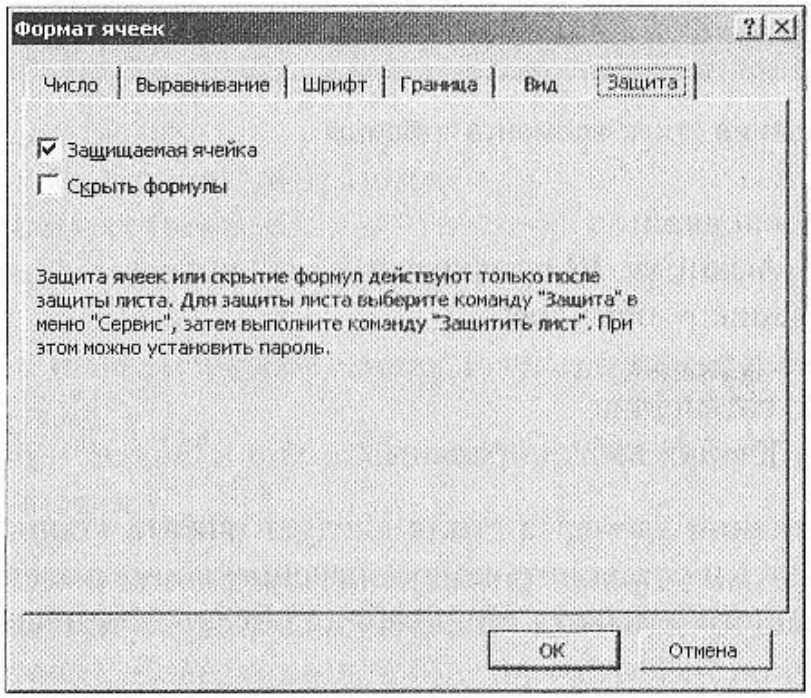


Рис. 5.11. Формат ячеек: вкладка «Защита»

Интересно применение условного форматирования (рис. 5.12). В этом случае оформление осуществляется только при выполнении заданного условия для выделенных ячеек таблицы. Например, при значении больше или равном 0 — полужирный курсивный шрифт, рамка и заливка.

Команда **Стиль** из меню **Формат** позволяет изменить во всей рабочей книге один или несколько элементов стиля (рис. 5.13):

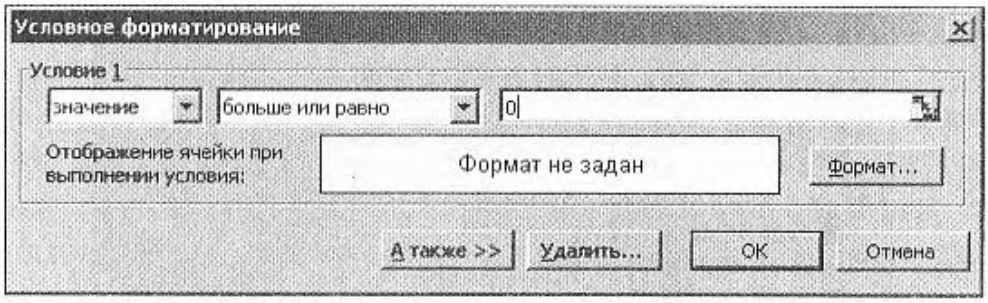


Рис. 5.12. Условное форматирование

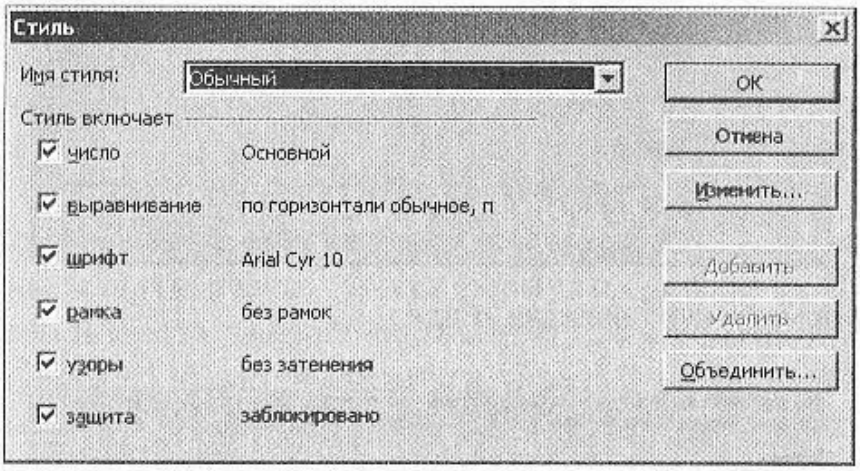


Рис. 5.13. Задание стиля элемента таблицы

- число — основной;
- шрифт — Arial Cyr, 10 пунктов;
- выравнивание — обычное;
- рамка — без рамки;
- вид — без затенения;
- защита — ячейки заблокированы.

Для изменения элемента стиля следует нажать кнопку **Изменить** и затем выбрать и установить новое значение данного элемента. После этого для данного элемента все в таблице будет изменено. Например, можно:

- изменить шрифт на Courier, девять пунктов, синего цвета, жирный;
- изменить защиту — не заблокировано;
- изменить узоры — светлый узор.

Построение графиков и диаграмм выполняется с помощью специальной программы **Мастер диаграмм**, запустить которую можно, щелкнув соответствующую пиктограмму либо выполнив команду меню **Вставка**, **Диаграмма** (рис. 5.14).

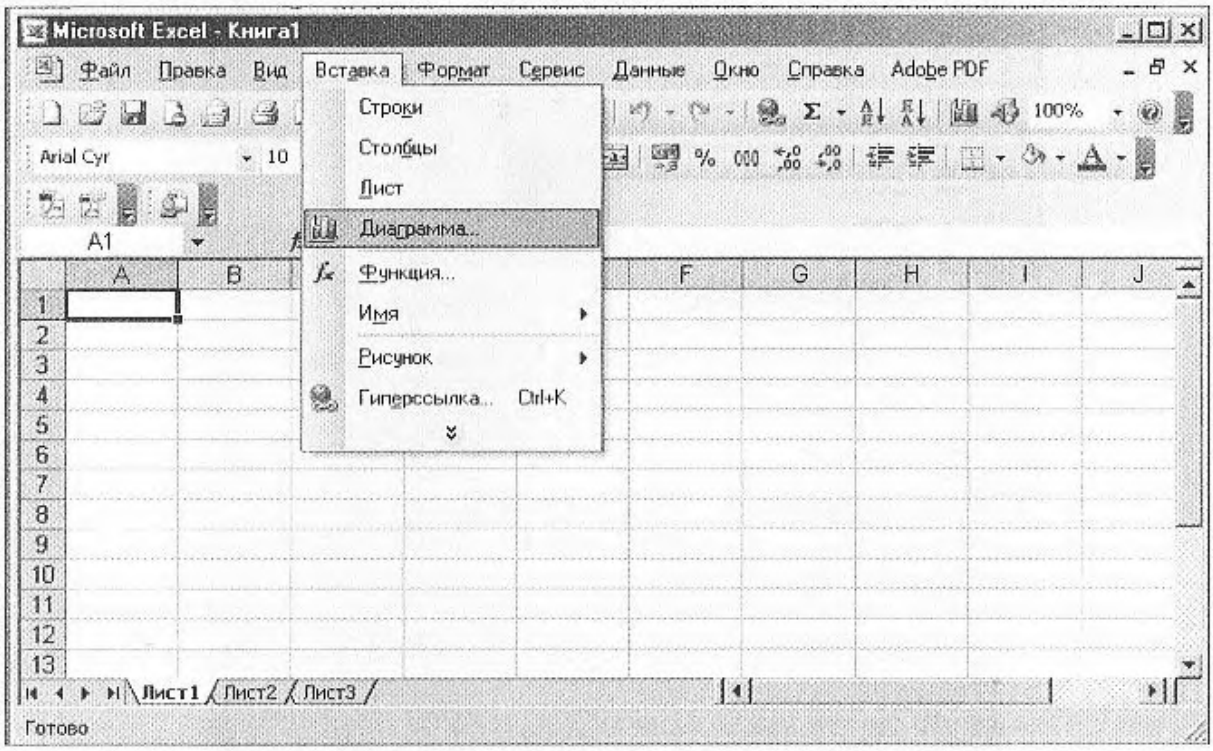


Рис. 5.14. Вызов Мастера диаграмм

Диаграмма включает в себя следующие элементы:

- область диаграммы;
- заголовок (название) диаграммы;
- область данных (чертеж);
- серии (ряды) данных — характеризуются специальной формулой ряда;
- легенда — набор маркеров, линий и текста, характеризующий ряд;
- оси X, Y и Z;
- категории — надписи под разметками деления оси X;
- подписи к осям.

Для построения графика (диаграммы) следует выделить область числовых значений вместе с категориями и легендами; вызвать команду меню **Вставка**, **Диаграмма** или **Мастер диаграмм** по пиктограмме.

Мастер диаграмм предусматривает четыре шага. На первом шаге выбирается *тип диаграммы* (стандартный, нестандартный) (рис. 5.15). На втором шаге определяется *источник данных* (диапазон и ряд) (рис. 5.16). На третьем шаге задаются *параметры диаграммы* (заголовки, оси, линии сетки, легенда, подписи данных) (рис. 5.17).

На четвертом шаге задается *размещение диаграммы* (на отдельном листе, на текущем листе) (рис. 5.18).

Для редактирования диаграмм предварительно следует войти в режим редактирования: перейти на лист **Диаграмма**, если она построена

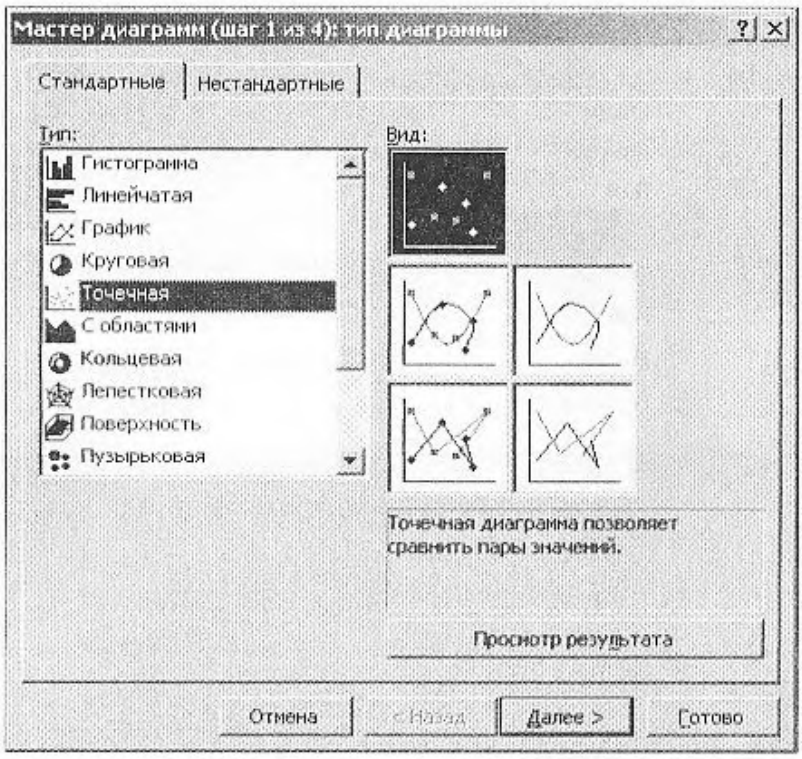


Рис. 5.15. Мастер диаграмм: шаг 1

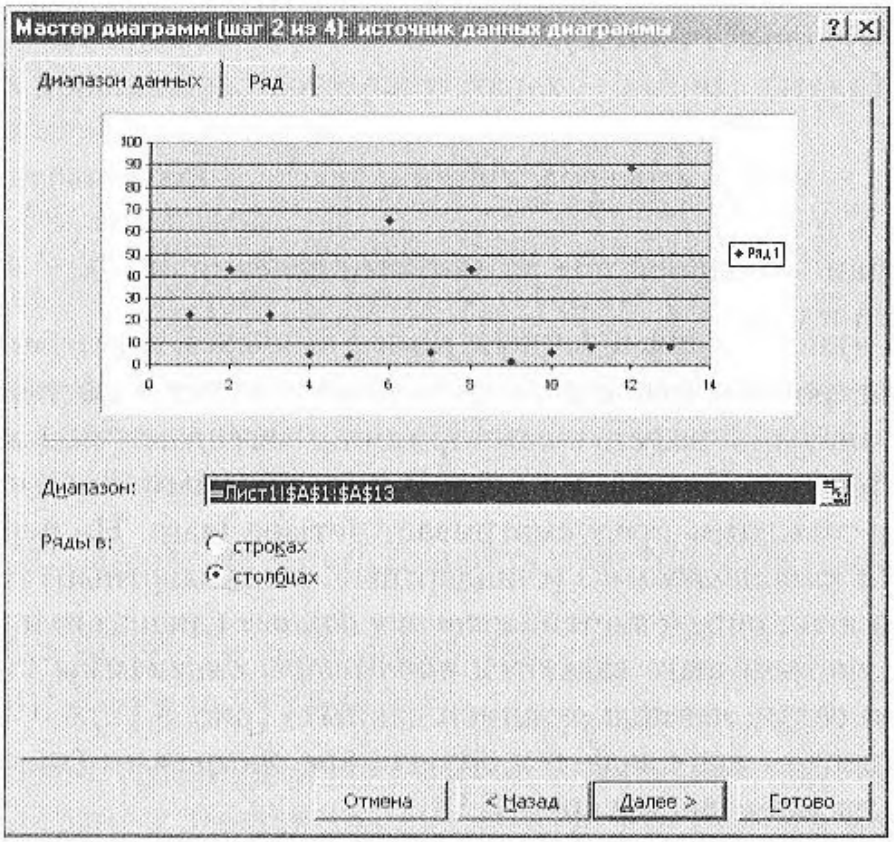


Рис. 5.16. Мастер диаграмм: шаг 2

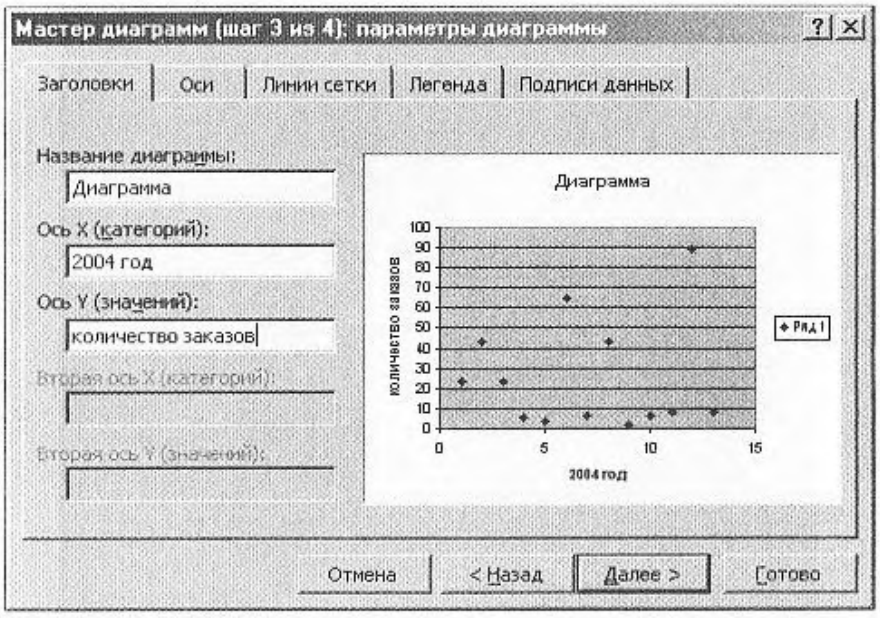


Рис. 5.17. Мастер диаграмм: шаг 3

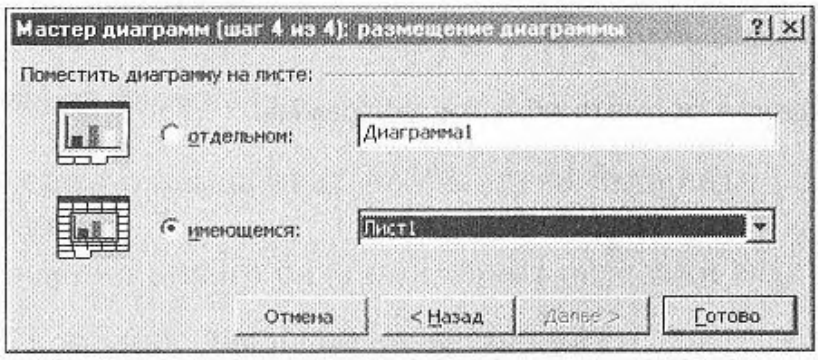


Рис. 5.18. Мастер диаграмм: шаг 4

на новом листе, или выделить диаграмму, если она построена на данном листе. Редактировать можно любой элемент диаграммы, который предварительно нужно выделить (отмечается маленькими квадратиками). Далее двойным щелчком либо соответствующей командой в меню **Формат** следует вызвать диалоговое окно изменения настроек.

Для названия диаграммы, подписей к осям и легенды можно изменить шрифт, свойства, вид. На вкладке **Вид** устанавливаются заливка области, рамка и ее стиль, тень (рис. 5.19).

Для рядов диаграммы можно настроить вид (цвет ряда и маркеры), установить еще одну вспомогательную ось, вывести метки данных, назначить новое имя ряда и новые значения по осям X и Y (рис. 5.20).

Категории форматируются вместе с осью X. Вкладка **Шрифт** позволяет выбирать нужный шрифт, вкладка **Число** — формат числа. Можно поменять также шкалу и вид оси (рис. 5.21). Для оси Y есть возможность изменить шкалу, вид, а также выбрать формат числа, шрифта и выравнивание.

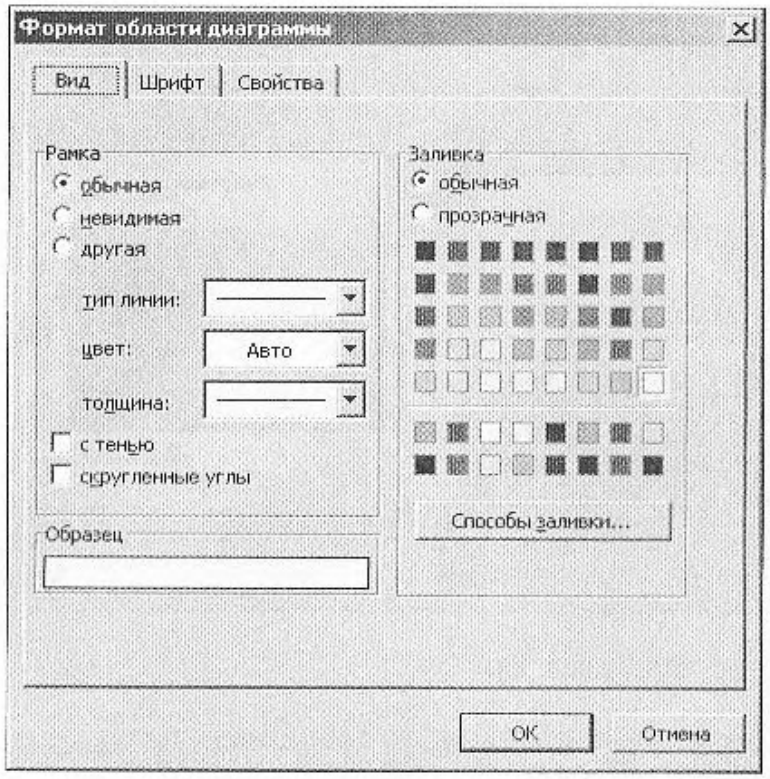


Рис. 5.19. Задание формата области диаграммы

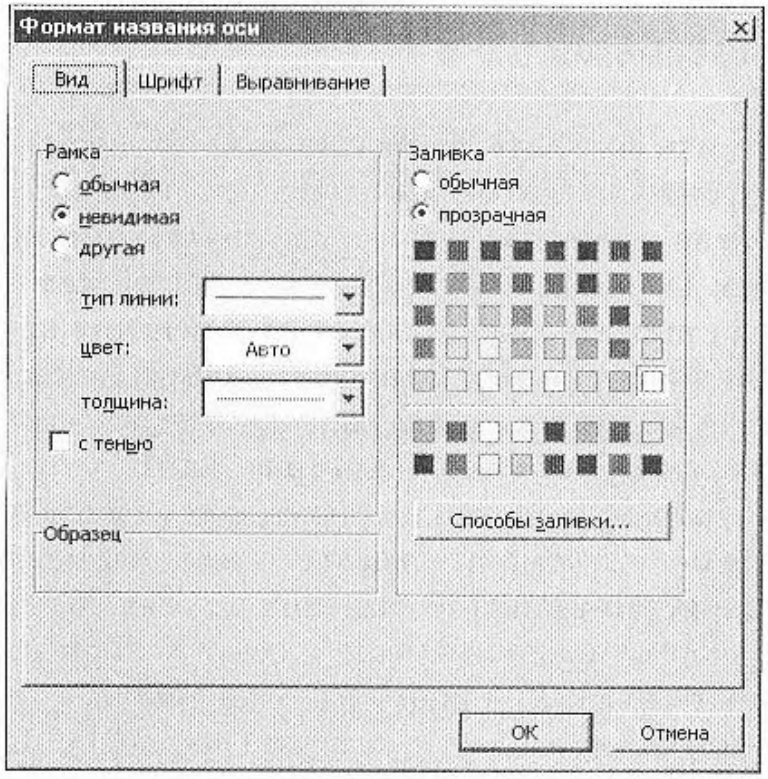


Рис. 5.20. Задание формата названия оси

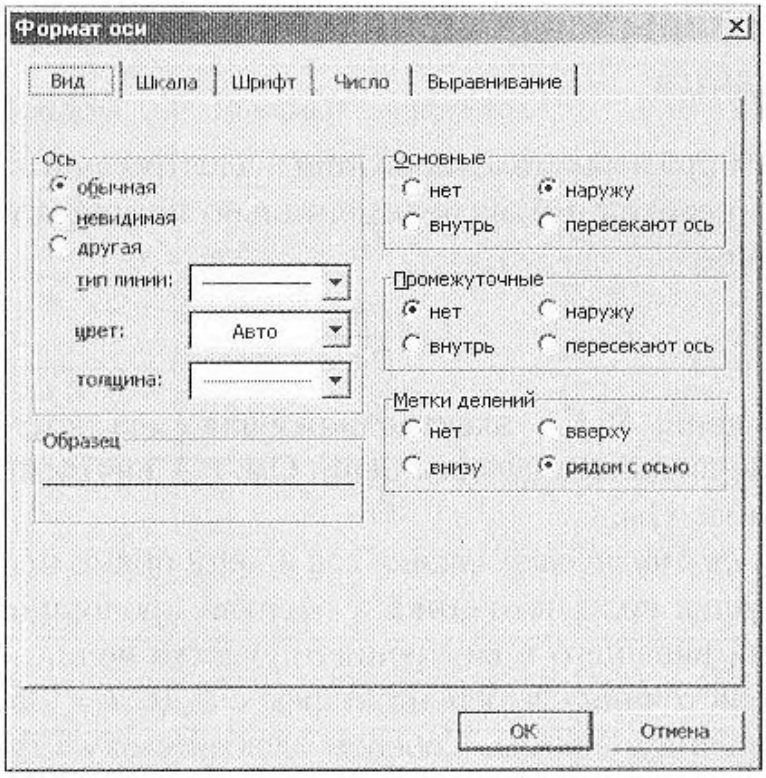


Рис. 5.21. Задание формата оси

Для удаления ряда данных необходимо выделить ряд и нажать клавишу **Delete**.

Для добавления нового ряда или новых значений можно:

- выделить в таблице числовые значения вместе с легендой ряда (для нового ряда) или без нее и скопировать их на область диаграммы;
- находясь в режиме редактирования диаграммы, вызвать команду **Диаграмма**, **Исходные данные**, вкладка **Ряд**, кнопка **Добавить**, а затем показать в таблице область добавляемых данных.

? Вопросы и задания

1. Назовите способы моделирования таблицы.
2. Как защитить от изменений лист, ячейку, группу ячеек?
3. Что можно изменить командой **Стиль**?
4. Назовите четыре шага **Мастера диаграмм**.
5. Сравните меню двух разных табличных сред (цв. вкл., рис. 5, 6).

П Проектное задание

Составьте таблицу учета оплаты коммунальных услуг за год. Наименование столбцов: **Дата оплаты**, **Вид коммунального платежа**, **Всего начислено к оплате**, **Оплачено**, **Остаток**. Постройте график ваших платежей за коммунальные услуги, расположив на оси X даты оплаты. Пересчитайте по текущему курсу коммунальные платежи в другой валюте и отобразите это на диаграмме.

5.3. Примеры моделирования в электронной таблице

Рассмотрим примеры моделирования в электронной таблице. Предположим, что необходимо экспериментально подтвердить закон Ома для полной цепи:

$$I = \frac{E}{R + r},$$

где I — ток в цепи, А; E — электродвижущая сила источника, В; R — сопротивление внешнего участка цепи, Ом; r — внутреннее сопротивление источника, Ом.

Напомним *формулировку закона*: ток в цепи прямо пропорционален электродвижущей силе источника и обратно пропорционален сумме сопротивлений внешнего и внутреннего участка цепи.

В качестве источника выберем батарею с ЭДС $E = 7{,}5$ В и внутренним сопротивлением 0,5 Ом. Соберем электрическую схему для проведения эксперимента (рис. 5.22).

Амперметр — это прибор, показывающий ток в цепи. С помощью реостата, изменяя положение движка, можно получать разные величины R — сопротивления внешнего участка цепи. Будем задавать с помощью реостата внешние сопротивления от 1 до 7 Ом с шагом 0,5. При каждом значении сопротивления смотрим показания амперметра и данные заносим в таблицу (табл. 5.4).

Некоторое расхождение экспериментальных и расчетных величин тока объясняется точностью измерения прибора. С помощью **Мастера диаграмм** построим график зависимости тока от сопротивления на основании полученной таблицы (рис. 5.23).

Рассмотрим другой пример. Роль статистики в жизни общества, да и каждого отдельного человека, трудно переоценить. Именно на основании статистических данных определяются такие стратегические показатели экономики страны, как валовой внутренний продукт (ВВП), динамика экономического роста, размер потребительской корзины,

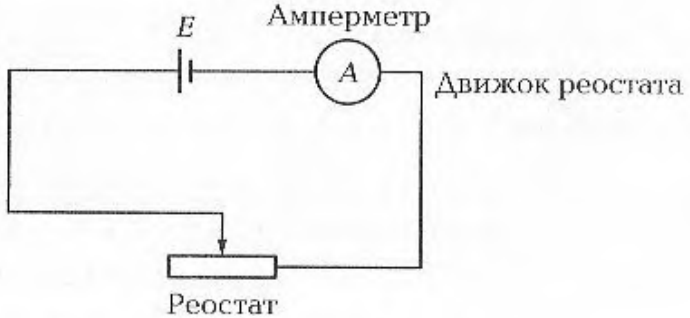


Рис. 5.22. Пример сборки электрической схемы

Таблица 5.4

Внешнее сопротивление, Ом	Показание амперметра, А	Расчетная величина тока, А
1	5	5
1,5	3,75	3,75
2	3	3
2,5	2,5	2,5
3	2,14	2,143
3,5	2,88	1,875
4	1,67	1,667
4,5	1,5	1,500
5	1,37	1,364
5,5	1,25	1,250
6	1,15	1,154
6,5	1,07	1,071
7	1	1,000

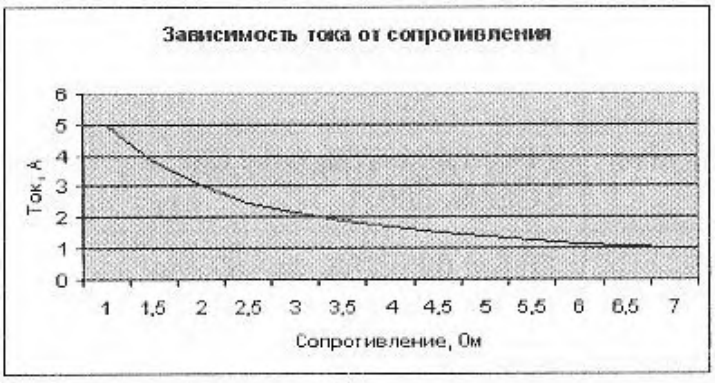


Рис. 5.23. График зависимости силы тока от сопротивления

средний прожиточный минимум человека и т.д. Статистика является не менее важным инструментом и для экономических структур — предприятий, фирм, организаций и т.д. Именно по статистическим показателям можно судить об эффективности финансово-хозяйственной деятельности предприятия, его рентабельности, оценивать вклад каждого подразделения, да и отдельного работника в общее дело. Широко применяется статистика в здравоохранении, образовании, правоохранительных органах, спорте, бизнесе, бухгалтерском учете и во многих других областях. Важностью статистики для жизни общества объясняется наличие в нашей стране развитой системы органов государственной статистики, от Росстата до территориальных статистических управлений и отделов. Очевидно, что любой статистический анализ возможен только на основании учетной информации, и чем выше достоверность этой информации, тем более качественные результаты будут получены. Сама статистика является серьезной наукой, широко использующей различные численные методы, которые в первую очередь основаны на теории вероятностей. Однако ее серьезное изучение осуществляется в специализированных вузах, поэтому ограничимся относительно простым примером.

Предположим, что мы работаем в компании, которая специализируется на оптовой и розничной продаже мониторов для компьютеров. Прайс-лист на эту продукцию, в котором указаны оптовые цены, которые действуют при покупке более пяти мониторов одновременно, и розничные цены представлены в табл. 5.5. В этом прайс-листе указаны: размер по диагонали в дюймах (15, 17, 19), марка монитора и приведены некоторые технические характеристики (TFT — жидкокристаллический экран, 1 024×786 — разрешающая способность, TCO99 — соответствие международному санитарно-гигиеническому стандарту и т.д.) и цены.

Таблица 5.5

Модель	Описание	Цена, у.е.	
		оптовая	розничная
15 Hansol H520 silver	TFT, 1 024×768, 140/120, TCO99, 300:1	315	320
15 Hyundai Q15	TFT, 1 024×786, 44—87 Гц, 31—69 кГц TCO99, 300:1	350	355
15 Samsung 151S	TFT, 1 024×768, 140/120, TCO99, 300:1	320	325

Окончание табл. 5.5

Модель	Описание	Цена, у.е.	
		оптовая	розничная
15 LG Flatron 566 LE	TFT, 1 024×768, 140/120, TCO99, 300:1	333	340
17 Samsung 171S	TFT, 1 280×1 024, 160/120, TCO99, 350:1	530	535
17 LG Flatron 786 LS	TFT, 1 280×1 024, 160/120, TCO99, 350:1	528	535
15 Hyundai V570	1 280×1 024@60, 0,28, TCO99	127	130
17 Hyundai V770	1 024×768@85, 0,28, TCO99	141	143
17 Hyundai Q770, F770	1 280×1 024@75, 0,20, TCO99, FLAT	169	172
17 Hyundai F790	1 600×1 280@75, 0,20, TCO99, FLAT	186	190
17 Hyundai Q790	1 600×1 280@75, 0,20, TCO99, FLAT	198	202
17 CTX PR711FL	1 600×1 200@75, 0,24, TCO99, FD Trinitron USB	255	260
17 LG F700B	1 280×1 024@60, 0,25, TCO99, Flatron	185	187
17 LG F700P	1 600×1 280@75, 0,24, TCO99, USB Flatron	215	218
17 LG 775FT	1 280×1 024@60, 0,25, TCO99, Flatron	185	187
17 Samsung 763MB	1 280×1 024@65, 0,24, TCO99, DF	179	183
17 Samsung 757NF	1 920×1 440@64, 0,20, TCO99, NF	235	238
19 Hyundai Q910	1 600×1 280@85, 0,20, TCO99, FLAT	300	305
19 HYAMA MA901U	1 480×1 024@90, Diamond-tron, TCO99, NF	370	375

Таблица 5.6

Марка	Цена оптовая, у.е.	Цена розничная, у.е.	Продано оптом, шт.	Доход от оптовых продаж, у.е.	Продано в розницу, шт.	Доход от розничных продаж, у.е.	Продано всего	Общий доход, у.е.	Объем оптовых продаж, %	Доход от оптовых продаж, %
15 Hansol H520 silver	315	320	15	4725	21	6720	36	11445	41,67	41,3
15 Hyundai Q15	350	355	23	8050	23	8165	46	16215	50	49,6
15 Samsung 151S	320	325	12	3840	14	4550	26	8390	46,15	45,8
15 LG Flatron 566 LE	333	340	11	3663	15	5100	26	8763	42,31	41,8
17 Samsung 171S	530	535	8	4240	12	6420	20	10660	40	39,8
17 LG Flatron 786 LS	528	535	6	3168	0	0	6	3168	100	100
15 Hyundai V570	127	130	10	1270	27	3510	37	4780	27,03	26,6
17 Hyundai V770	141	143	21	2961	22	3146	43	6107	48,84	48,5
17 Hyundai Q770	169	172	11	1859	23	3956	34	5815	32,35	32
17 Hyundai F790	186	190	12	2232	11	2090	23	4322	52,17	51,6
17 Hyundai Q790	198	202	23	4554	6	1212	29	5766	79,31	79

Окончание табл. 5.6

Марка	Цена оптовая, у.е.	Цена розничная, у.е.	Продано оптом, шт.	Доход от оптовых продаж, у.е.	Продано в розницу, шт.	Доход от розничных продаж, у.е.	Продано всего	Общий доход, у.е.	Объем оптовых продаж, %	Доход от оптовых продаж, %
17 CTX PR711FL	255	260	20	5100	21	5460	41	10560	48,78	48,3
17 LG F700B	185	187	0	0	10	1870	10	1870	0	0
17 LG F700P	215	218	17	3655	4	872	21	4527	80,95	80,7
17 LG 775FT	185	187	14	2590	3	561	17	3151	82,35	82,2
17 Samsung 763MB	179	183	18	3222	6	1098	24	4320	75	74,6
17 Samsung 757NF	235	238	27	6345	3	714	30	7059	90	89,9
19 Hyundai Q910	300	305	6	1800	5	1525	11	3325	54,55	54,1
19 IIYAMA MA901U	370	375	11	4070	0	0	11	4070	100	100
	Итого		265	67344	226	56969	491	124313	—	—
							Средний процент		57,45	57,1

Отчет должен содержать количество мониторов, проданных оптом и в розницу в конкретном месяце (например, мае), соответствующие суммы доходов и должен позволить сделать вывод о том, какие имен-

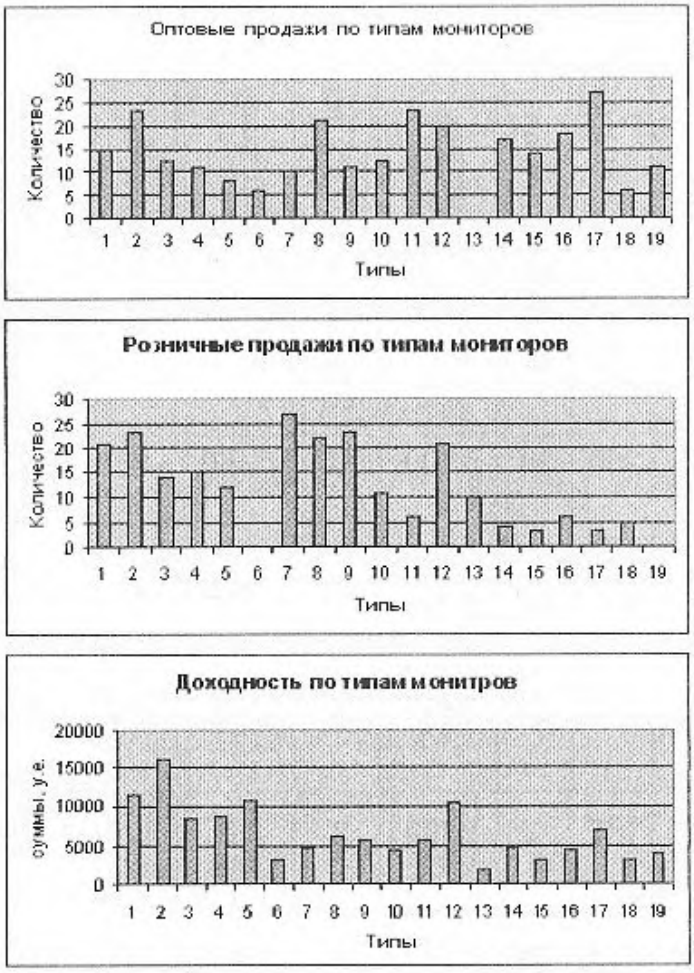


Рис. 5.24. Графические материалы для иллюстрации отчета

но мониторы пользовались бóльшим спросом и принесли наибольший доход (табл. 5.6). Полученные данные этого статистического анализа можно рассматривать как результаты маркетингового исследования, которые помогут внести соответствующие коррективы в политику продаж компании.

Отчет целесообразно проиллюстрировать графическими материалами: графиками и диаграммами, которые придадут ему большую наглядность (рис. 5.24).

Из статистического отчета можно определить, какие мониторы больше пользуются спросом у оптовых покупателей, а какие — у розничных, и в каком соотношении. На основании информации с учетом факторов развития рынка и прогнозируемой динамики цен можно вырабатывать стратегию продаж на следующий период.

Проектное задание

Выполните моделирование и оформление таблицы для своей курсовой работы в соответствии с профилем обучения. Воспользуйтесь примерами данного параграфа.

5.4. База данных как модель информационной структуры

Модель представления данных очень важна для понимания работы с информационными структурами — базами данных. Современная наука рассматривает три основные модели организации данных: иерархическую, сетевую и реляционную. *Иерархическую модель* организации данных мы рассмотрели ранее на примере файловой системы и выяснили, как она реализована на диске с помощью каталога и таблицы файлов. С *сетевой моделью* мы познакомимся позже на примере компьютерных сетей.

Реляционная модель описывает, какие данные могут храниться в БД, а также способы управления такими данными. В упрощенном виде основная идея реляционной модели состоит в том, что данные должны храниться только в таблицах, каждый набор данных может иметь свою таблицу, в том числе вложенную, но все эти таблицы должны иметь связи друг с другом. Таким образом, такие наборы таблиц могут собираться в зависимости от способа представления данных по запросу из базы в совершенно произвольную структуру. Например, если рассматривать БД сотрудников большого предприятия, то по запросу «представить сотрудников по отделам с указанием детей этих сотрудников и зоны интересов этих детей» (в целях организации досугового центра для детей сотрудников) получится табл. 5.7.

Таблица 5.7

Отдел	Сотрудник	Дети сотрудников (интересы)		
Склад	Петухов И.И.	Таня	Моделирование	
		Коля	Литература	Видео
		Петя	Компьютерные игры	
		Саша	Спорт	
Склад	Синицын П.А.	Дима	Электроника	
	Дроздов А.С.	Алена	Программирование Моделирование	
		Нина	Литература	
		Оксана	Прикладное творчество	
Бухгалтерия	...	...		

Реляционная модель основана на теории множеств и математической логике. Такой фундамент обеспечивает математическую строгость реляционной модели данных.

Классическая реляционная модель данных требует, чтобы данные хранились в так называемых *плоских таблицах*. В упрощенном виде плоской является таблица, каждая ячейка которой может быть однозначно идентифицирована указанием строки и столбца таблицы — адресом ячейки. Кроме того, в одном столбце все ячейки должны содержать данные одного простого типа.

На основе реляционной модели данных были разработаны различные языки для написания и управления реляционными базами данных, например SEQUEL, SQL, QUEL и др. Фактически промышленным стандартом в настоящее время стал язык SQL (Structured Query Language — язык структурированных запросов).

Взаимосвязь реляционной модели данных, стандарта языка SQL и различных его реализаций можно условно изобразить в виде пирамиды, изображенной на рис. 5.25.

Каждый более высокий уровень основывается на понятиях, определенных на более низком уровне. На каждом из уровней используется своя терминология. Например, на уровне теории множеств говорят «множество», «подмножество», «кортеж». На уровне реляционной модели используют термины «домен», «отношение», «кортеж». На уровне стандарта SQL и конкретных реализаций применяют термины «тип данных», «таблица», «строка таблицы», хотя все они отражают практически одно и то же.

Основы реляционной модели данных были впервые изложены в статье Эдгара Кодда в 1970 г. Эта работа послужила стимулом для написания множества статей и книг, в которых реляционная модель получила дальнейшее развитие. Наиболее распространенная трактовка реляционной модели данных принадлежит Кристоферу Дейту. Согласно Дейту, реляционная модель состоит из трех частей: структурной, целостной и манипуляционной.

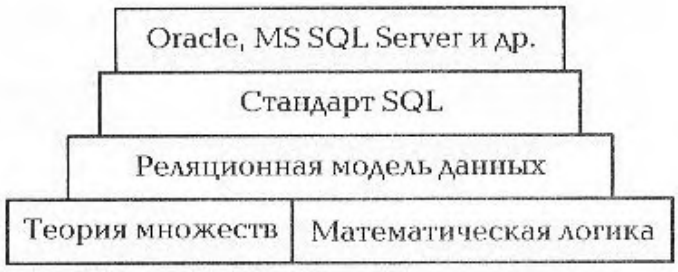


Рис. 5.25. Схематичное отображение взаимосвязи реляционной модели данных, стандарта языка SQL и его реализаций

Структурная часть описывает, какие объекты рассматриваются реляционной моделью. Постулируется, что единственной структурой данных, используемой в реляционной модели, являются парные отношения. *Целостная часть* описывает ограничения специального вида, которые должны выполняться для любых отношений в любых реляционных БД. Это целостность ключей. *Манипуляционная часть* описывает два эквивалентных способа манипулирования реляционными данными — реляционную алгебру и реляционное исчисление. Конечно, должны быть описаны действия, которые можно производить с данными как с единым целым, например, данные числового типа можно складывать, для строк возможна операция конкатенации (объединения) и т. д.

В реляционной модели данных с понятием «тип данных» тесно связано понятие *домена*, которое можно считать уточнением типа данных. Домен — семантическое (смысловое) понятие. Его можно рассматривать как подмножество значений некоторого типа данных, имеющих определенный смысл. Домен характеризуется следующими свойствами:

- имеет уникальное имя (в пределах БД);
- определен на некотором простом типе данных или на другом домене;
- может иметь некоторое логическое условие, позволяющее описать подмножество данных, допустимых для данного домена;
- несет определенную смысловую нагрузку.

Если тип данных можно считать множеством всех возможных значений данного типа, то домен напоминает подмножество в этом множестве. Отличие домена от подмножества состоит именно в том, что домен отражает семантику, определенную предметной областью. Может быть несколько доменов, совпадающих по типу данных, но несущих различный смысл. Например, домены «Вес детали» и «Имеющееся количество» можно одинаково описать как множество неотрицательных целых чисел, но смысл этих доменов будет различным, и это будут различные домены.

Основное значение доменов состоит в том, что домены ограничивают сравнение данных в базе. Некорректно с логической точки зрения, сравнивать значения из различных доменов, даже если они имеют одинаковый тип данных. Запрос «выдать список всех деталей, у которых масса детали больше имеющегося количества» не соответствует смыслу понятий «количество» и «масса» и является некорректным, хотя сравнение чисел — допустимая операция. Система даст ответ, но вероятно он будет бессмысленным.

Не всегда очевидно, как задать логическое условие, ограничивающее возможные значения домена. Ясно, что строки, являющиеся фамилиями, не должны начинаться с цифр, служебных символов, мягкого знака и т.д. Но вот является ли допустимой фамилия «Аааааааааббб»? Очевидно, нет! Система не выдаст ошибки, но с точки зрения смысловой нагрузки такая фамилия внесет ошибку в БД. Подобная правильная по типу, но неверная по смыслу информация является «мусором».

Фундаментальным понятием реляционной модели данных является «отношение». *Атрибут отношения* есть пара следующего вида:

<Имя_атрибута : Имя_домена>

Имена атрибутов должны быть уникальны в пределах отношения. Часто имена атрибутов отношения совпадают с именами соответствующих доменов.

Отношение, определенное на множестве доменов (не обязательно различных), содержит две части: заголовок и тело.

Заголовок отношения включает в себя фиксированное количество атрибутов отношения. Набор заголовков отношений, входящих в БД, называется *схемой реляционной БД*.

Тело отношения содержит множество кортежей отношения. Каждый *кортеж отношения* представляет собой множество пар вида

<Имя_атрибута : Значение_атрибута>

Количество кортежей отношения называется *мощностью отношения*. Число атрибутов в отношении называют *степенью* отношения.

Рассмотрим отношение «Сотрудники», заданное на доменах «Номер_сотрудника», «Фамилия», «Зарплата», «Номер_отдела». Все домены различны, поэтому имена атрибутов отношения удобно назвать так же, как и соответствующие домены. Заголовок отношения имеет следующий вид:

Сотрудники (Номер_сотрудника, Фамилия, Зарплата, Номер_отдела)

Заголовок отношения «Сотрудники» содержит четыре атрибута: «Номер_сотрудника», «Фамилия», «Зарплата», «Номер_отдела».

Пусть в данный момент тело отношения содержит три кортежа (мощность отношения равна 3):

(1, Петухов, 5000, 2)

(2, Синицын, 7000, 1)

(3, Дроздов, 8500, 2)

Каждый кортеж включает в себя четыре пары. Например, для первого кортежа:

Номер_сотрудника: 1.

Фамилия: Петухов.

Зарплата: 5000.

Номер_отдела: 2.

Такое отношение «Сотрудники» представляется в виде табл. 5.8.

Таблица 5.8

Номер_сотрудника	Фамилия	Зарплата	Номер_отдела
1	Петухов	5 000	2
2	Синицын	7 000	1
3	Дроздов	8 500	2

Таким образом, *реляционной базой данных* называется набор отношений.

Хотя любое отношение можно изобразить в виде таблицы, нужно четко понимать, что отношения сами не являются таблицами, а лишь представляются в такой форме. Термины, которыми оперирует реляционная модель БД, имеют соответствующие табличные синонимы (табл. 5.9).

Основной особенностью компьютерных реляционных БД является высокая скорость доступа к данным. Поскольку каждый элемент таблицы — клетка (ячейка) — имеет свой адрес, то к ней можно обратиться напрямую, минуя остальные элементы базы. Такой способ доступа называется *прямым*.

Возможность быстрого прямого доступа к данным была не сразу использована для работы с БД на компьютерах. Развитие моделей табличного представления данных сопровождалось поиском технологий аппаратной реализации устройств хранения информации, поскольку это влияло на скорость работы с большими объемами информации на компьютере. Высокая скорость обработки информации на компьютере тормозилась медленной работой устройств хранения информации. Первоначально в компьютерах использовались устройства с *последовательным доступом* к данным на внешних носителях информации. Именно таким способом читается информация, записанная на магнитной ленте кассетного магнитофона: просматриваются все записи, одна за другой, до тех пор, пока не будет найдена нужная (перемотка). Излишне говорить, что последовательный способ доступа к данным обеспечивает гораздо меньшую скорость. Примером компьютерных устройств, использующих последовательный способ доступа к данным, является магнитофон, который для компьютера известен как *стример* (рис. 5.26).

Принципиально новым стало создание устройств прямого доступа, в частности накопителей на магнитных дисках (см. рис. 2.19, *б*).

Таблица 5.9

Реляционный термин	Табличный термин
База данных	Набор таблиц
Схема базы данных	Набор заголовков таблиц
Отношение	Таблица
Заголовок отношения	Заголовок таблицы
Тело отношения	Тело таблицы
Атрибут отношения	Наименование столбца таблицы
Кортеж отношения	Строка таблицы
Степень (арность) отношения	Количество столбцов таблицы
Мощность отношения	Количество строк таблицы
Домены и типы данных	Типы данных в ячейках таблицы

Основной особенностью таких устройств является движение элемента считывания-записи информации (магнитной головки диска), который подводится непосредственно к тому месту на носителе, откуда необходимо прочитать или куда необходимо записать информа-

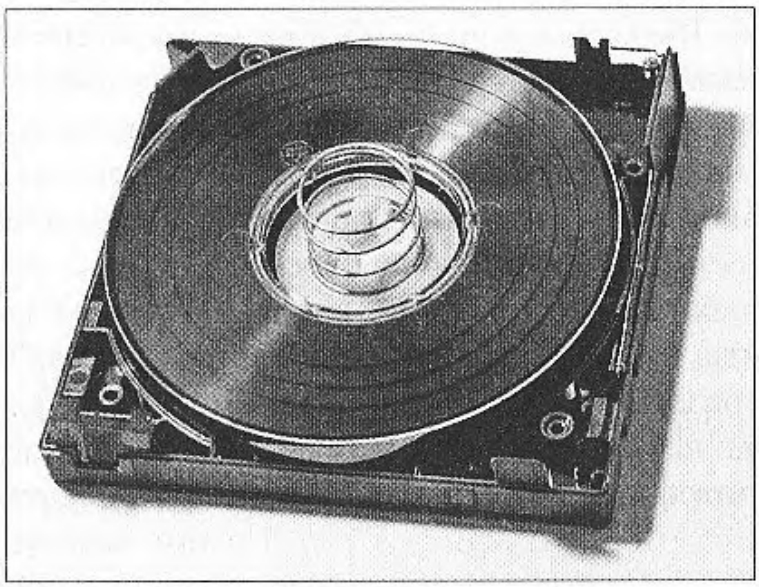


Рис. 5.26. Ленточный картридж для стримера

цию. Это и называется *прямым доступом к информации*. Естественно, что скорость доступа к информации в таких устройствах является достаточно высокой, что позволило развивать системы управления БД, которые играют в настоящее время важнейшую роль в новых информационных технологиях.

Вопросы и задания

1. Какие существуют структуры организации данных?
2. Что представляет собой иерархическая модель организации данных? Объясните на примере каталога файловой системы.
3. Перечислите основные понятия реляционной модели данных. Объясните их на примере.
4. Что представляет собой реляционная БД?
5. Назовите способы доступа к массивам инормации. Охарактеризуйте их.

5.5. Компьютерная база данных — система организации, хранения, доступа, обработки и поиска информации

Понятие «база данных» имеет достаточно широкий смысл. Для сравнения можно привести склад товаров, где товары лежат не вперемешку, а в рассортированном по наименованию виде и каждый на своей полке. Другое дело, насколько оптимально они рассортированы и расположены. Например, информация о том, какие именно товары есть на складе, каковы их основные характеристики и какое количество каждого есть в наличии, может рассматриваться в качестве складской БД.

Наиболее характерными информационными базами являются картотеки. Рассмотрим создание картотеки своих знакомых или партнеров по работе с использованием программы Windows «Адресная книга». В меню **Пуск** выберем **Программы**, **Стандартные**, а затем **Адресная книга**. Появится незаполненная табличная структура книги (рис. 5.27). Для создания контакта с меню этого окошка выполним **Файл**, **Создать контакт**.

В соответствии с вкладками можно классифицировать контакты как служебные, личные и т.д. Вы увидите, что в первой табличке появилась главная идентификационная запись с нашим знакомым. Выберем вкладку **Имя** и занесем данные — фамилию, имя, отчество, должность и псевдоним нашего знакомого. После заполнения поля нажмем клавишу **Добавить** (рис. 5.28). В итоге главная идентификационная запись в первом окне будет выглядеть так, как показано на рис. 5.29.

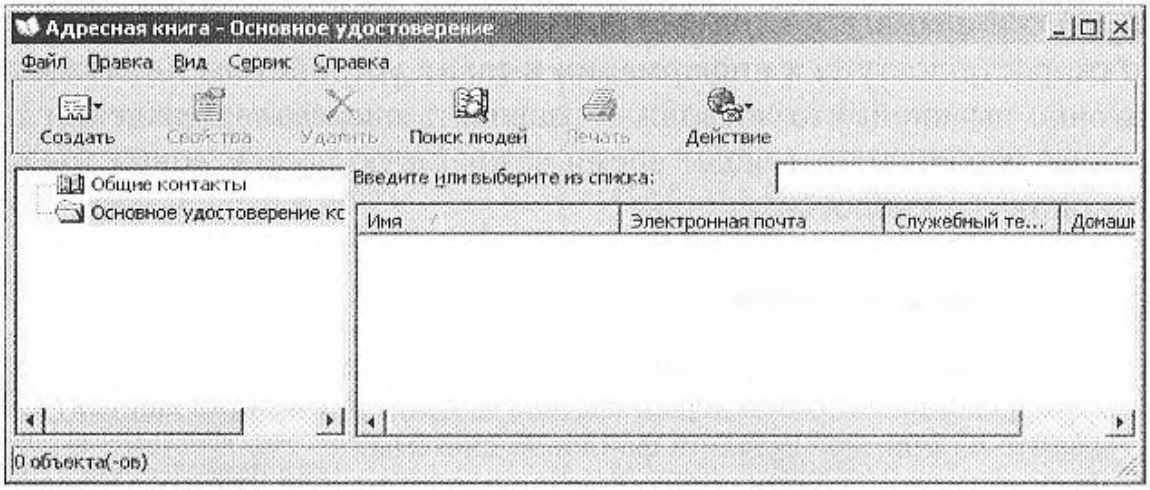


Рис. 5.27. Адресная книга

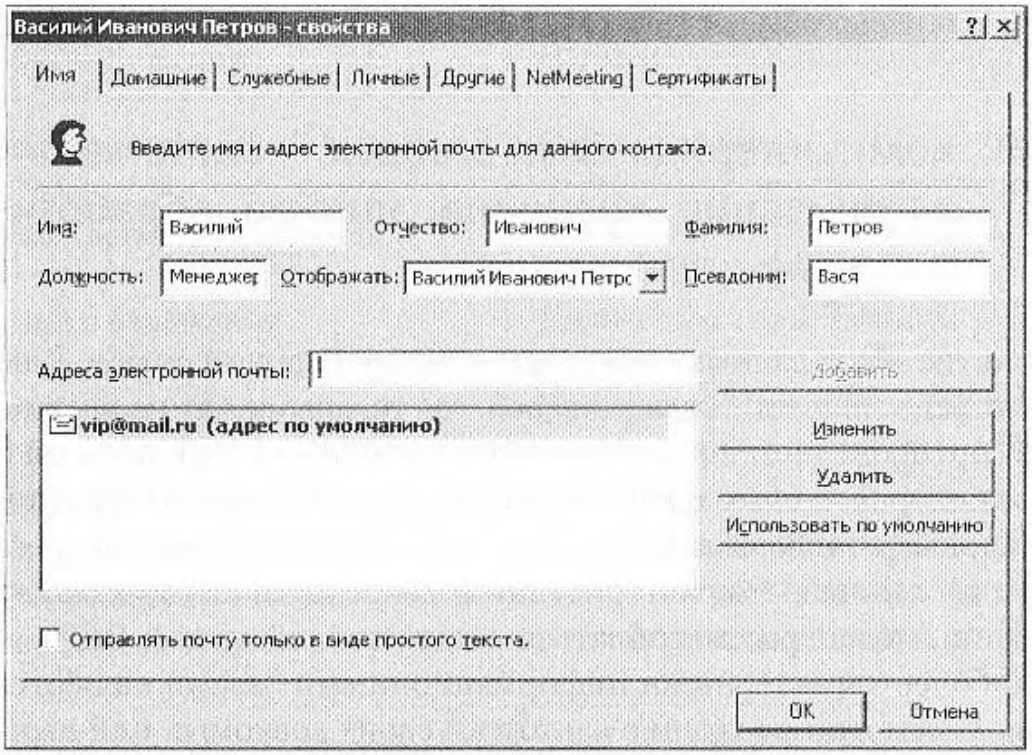


Рис. 5.28. Создание записи в адресной книге

Создав свою БД «Адресная книга», можно осуществлять поиск необходимой информации, задавая те или иные параметры поиска либо их определенные комбинации в команде **Поиск людей**.

В базах данных, использующих табличную структуру, необходимо различать объекты:

- *поле базы данных* — столбец таблицы;
- *запись в базе данных* — строка таблицы, содержащая информацию из различных полей (столбцов);
- *ключевое поле* (*ключ*) — одно из полей записи, однозначно идентифицирующее (определяющее) всю запись. В качестве ключа может

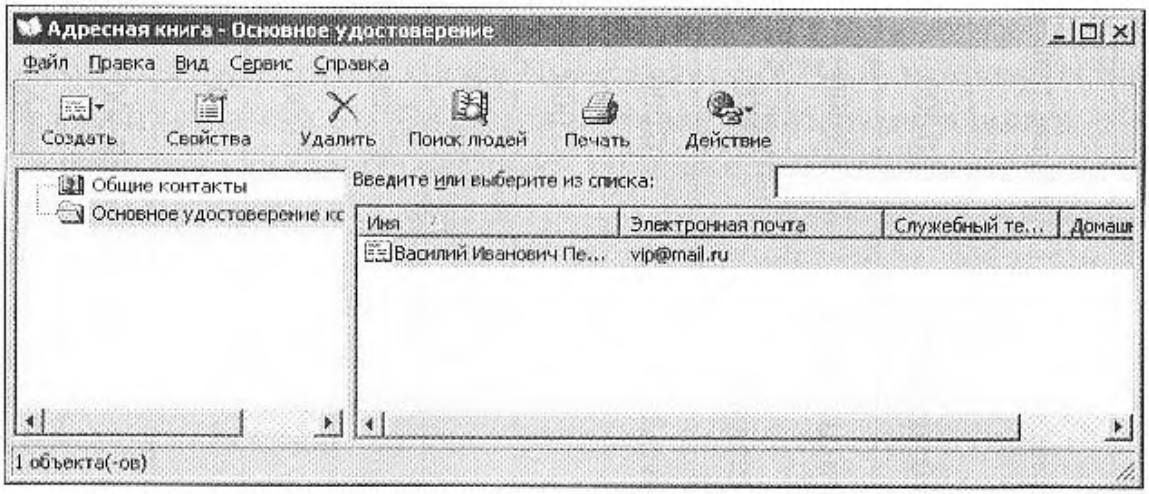


Рис. 5.29. Запись в адресной книге

использоваться фамилия, артикул товара, номер записи в реестре и т. п.

Поля могут быть разнотипными, в частности принадлежать к следующим типам данных:

- *счетчик* — является последовательностью номеров записей, которую мы изменить не можем;
- *текстовый* — содержит текст длиной до 255 символов;
- *числовой* — включает в себя числа;
- *дата/время* — содержит информацию о дате и времени;
- *денежный* — используется при финансовых расчетах и содержит денежный формат данных;
- *логический* (*булевский*) — содержит значения TRUE (истина) или FALSE (ложь);
- *гиперссылка* — содержит адрес какой-либо страницы в Интернете;

Каждый тип поля характеризуется набором свойств, например:

- *размер* — определяет максимальную длину (тестовые и числовые поля);
- *формат* — устанавливает формат представления данных;
- *обязательное поле* — указывает, что данное поле должно быть заполнено.

Рассмотрим пример БД по персоналу небольшой фирмы (табл. 5.10).

Типы данных в полях: **Номер** — счетчик; **Фамилия**, **Имя**, **Отчество**, **Должность** — текстовые, **Год рождения**, **Оклад** — числовые.

Для управления БД существуют специальные СУБД, которые можно сравнить с полностью автоматизированным складом, на котором достаточно запросить товар, и специальное устройство тут же предо-

Таблица 5.10

Номер	Фамилия	Имя	Отчество	Год рождения	Должность	Оклад, руб.
1	Дроздов	Петр	Васильевич	1964	Генеральный директор	20 000
2	Соколов	Игорь	Петрович	1962	Финансовый директор	18 000
3	Воронов	Сергей	Иванович	1966	Главный менеджер	15 000
4	Канарейкина	Анна	Ивановна	1970	Главный бухгалтер	16 000
5	Петухов	Василий	Олегович	1971	Менеджер	13 000
6	Синицын	Олег	Сергеевич	1968	Инженер	12 500
7	Галкин	Максим	Игоревич	1973	Менеджер	13 000
8	Кукушкина	Галина	Петровна	1975	Кладовщик	8 000
9	Соловьев	Сергей	Алексеевич	1974	Электрик	9 000
10	Сорокин	Федор	Федорович	1979	Экспедитор	7 500
11	Стрижев	Кирилл	Иванович	1980	Водитель	10 000

ставит его вам. Наиболее распространенной СУБД является Microsoft Access, хотя существуют и куда более мощные СУБД, например Oracle, Sybase, Interbase и др. **Системы управления базами данных** — это программы, которые обеспечивают создание, актуализацию, корректировку, поддержание БД, а также обработку этих данных и поиск. Если говорить о СУБД Access, то основным элементом ее является *окно базы данных*. На экране одновременно может быть много окон, так как Windows обеспечивает многооконный интерфейс, но активным в данный момент времени может быть только одно. Основными объектами БД Access являются:

- *таблицы* — базовый, основной объект БД, все остальные (вторичные) объекты создаются на его основе. Напомним, что строки (записи) содержат разнотипные поля, относящиеся к одному объекту хранения, а столбцы — однотипные поля для всех объектов хранения;
- *запросы* — осуществляют поиск информации по заданным условиям;
- *формы* — позволяют отображать и редактировать информацию из БД, могут содержать графическую информацию и другие объекты;
- *отчеты* — предназначены для вывода на принтер в удобном виде результатов обработки информации из БД;
- *макросы* — это фактически макрокоманды, позволяющие автоматизировать повторяющиеся или часто встречающиеся действия с информацией из БД;
- *программные модули* — дают возможность писать программы по обработке информации из БД.

Создать базу данных по персоналу гипотетической организации можно следующим образом:

1. Вызвать MS Access.

2. Выбрать команду **Файл**, **Создать базу данных** и присвоить созданной базе имя «Персонал».

3. В появившемся окне **Персонал: база данных** на вкладке **Таблицы** выбрать **Создание таблицы в режиме конструктора** (этот режим позволяет создавать и корректировать структуру таблицы).

4. В окне конструктора ввести названия полей (в столбце **Имя поля**), типы соответствующих данных с помощью раскрывающегося списка данных (в столбце **Тип данных**) и свойства полей (в нижней части окна). Для нашего примера следует указать семь полей: **Номер**, **Фамилия**, **Имя**, **Отчество**, **Год рождения**, **Должность**, **Оклад**.

5. Выбрать ключевое поле. Для этого вызвать контекстное меню нажатием правой клавиши мыши на нужном поле (в нашем примере — на поле **Номер**) и выбрать **Ключевое поле**.

6. Сохранить таблицу с помощью команды **Файл**, **Сохранить как...**, указав имя «Персонал». Закрыть окно **Персонал: таблица**.

7. Для просмотра созданной структуры в окне **Персонал: база данных** на вкладке **Таблицы** выбрать таблицу «Персонал» и нажать кнопку **Открыть**. Появится окно таблицы (рис. 5.30).

8. Ввести данные в созданную таблицу. В открытом окне **Персонал: таблица** слева от полей таблицы находится поле маркера. Маркер-стрелка указывает на активную запись (запись, с которой вы работаете в данный момент). Заполнить таблицу, определив значения всех полей для нескольких записей (рис. 5.31).

Форма отображает записи таблицы в удобном для пользователя виде. С ее помощью можно изменить расположение полей, улучшить дизайн, подключить графику и т. п. Фактически с помощью формы создается графический интерфейс доступа к БД, который может содержать такие элементы, как кнопки, переключатели, надписи и т. п. Формы создаются либо с помощью **Конструктора**, либо, что проще, с помощью **Мастера форм**.

Чтобы создать форму для БД «Персонал» с помощью **Мастера форм**, необходимо:

1. В окне **Персонал: база данных** на вкладке **Формы** выбрать **Создание формы с помощью мастера**.

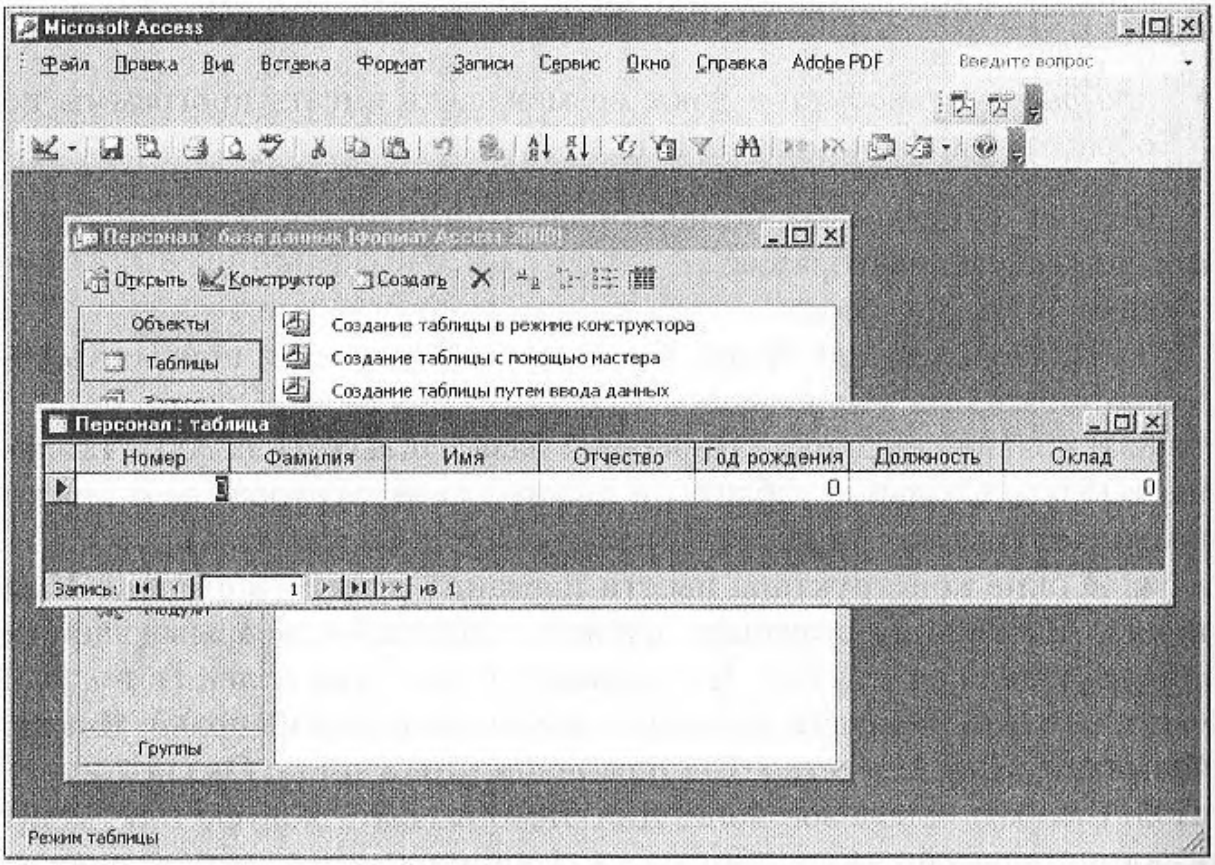


Рис. 5.30. Структура поля базы данных «Персонал»

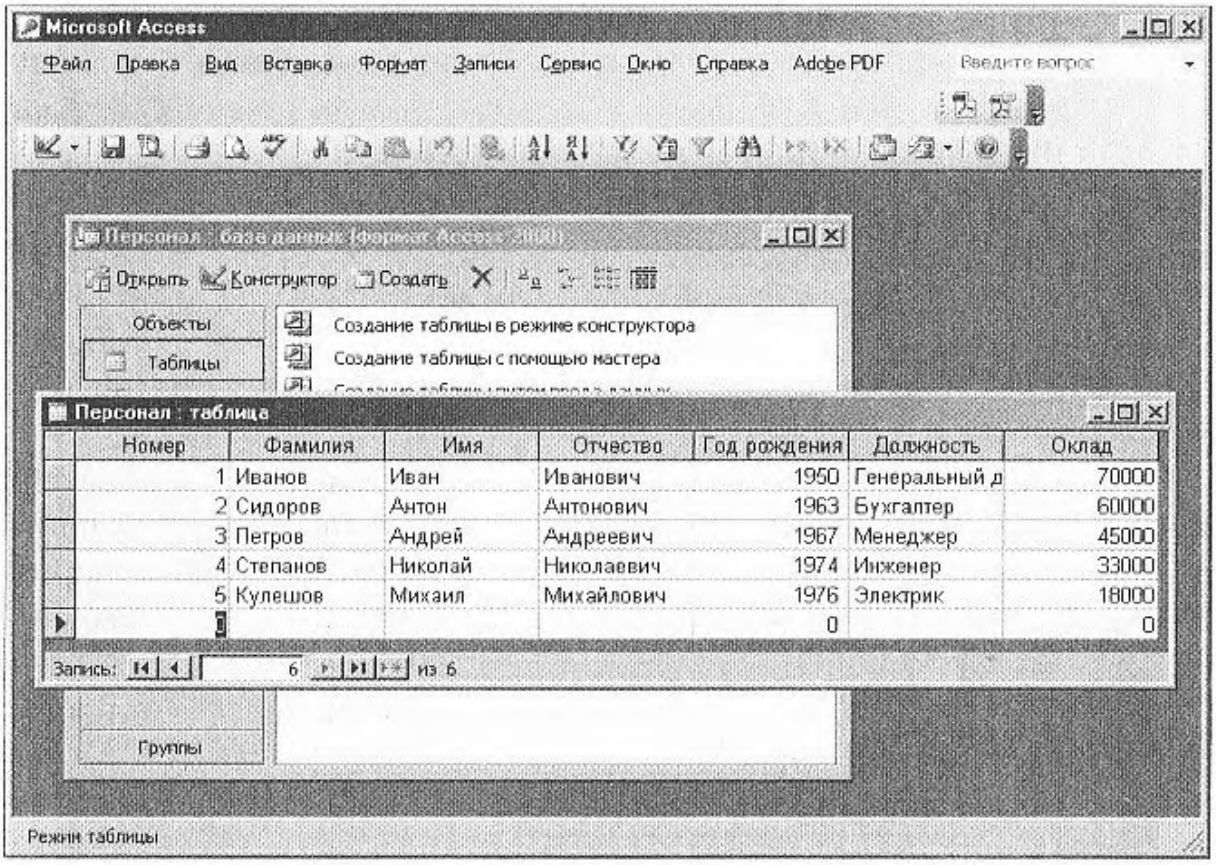


Рис. 5.31. База данных «Персонал»

2. В открывшемся окне **Создание форм** выбрать в области **Таблицы и запросы** таблицу «Персонал». Из области **Доступные поля** в область **Выбранные поля** перенести те поля выбранной таблицы, которые необходимо отобразить в форме (для нашего примера перенести все поля). Щелкнуть **Далее**.

3. На открывшейся панели с помощью переключателей выбрать способ размещения полей на форме (для нашего примера выбрать вариант **В один столбец**), затем щелкнуть **Готово**. В результате появится окно формы «Персонал», которое содержит имена полей формы и поля для ввода их значений, расположенных выбранным способом.

4. При необходимости изменить вид элементов формы (укрупнить надписи, изменить размер и т. д.). Для этого вызвать контекстное меню нажатием правой кнопки мыши на заголовке окна формы «Персонал» и выбрать **Конструктор форм**. Затем закрыть окно формы «Персонал».

5. Для просмотра формы в окне **Персонал: база данных** на вкладке **Формы** выбрать форму «Персонал» и нажать кнопку **Открыть**. Появится окно созданной формы. Перемещаться по записям можно с помощью кнопок в нижней части окна формы (рис. 5.32).

Поиск данных в таблице выполняется по следующему алгоритму:

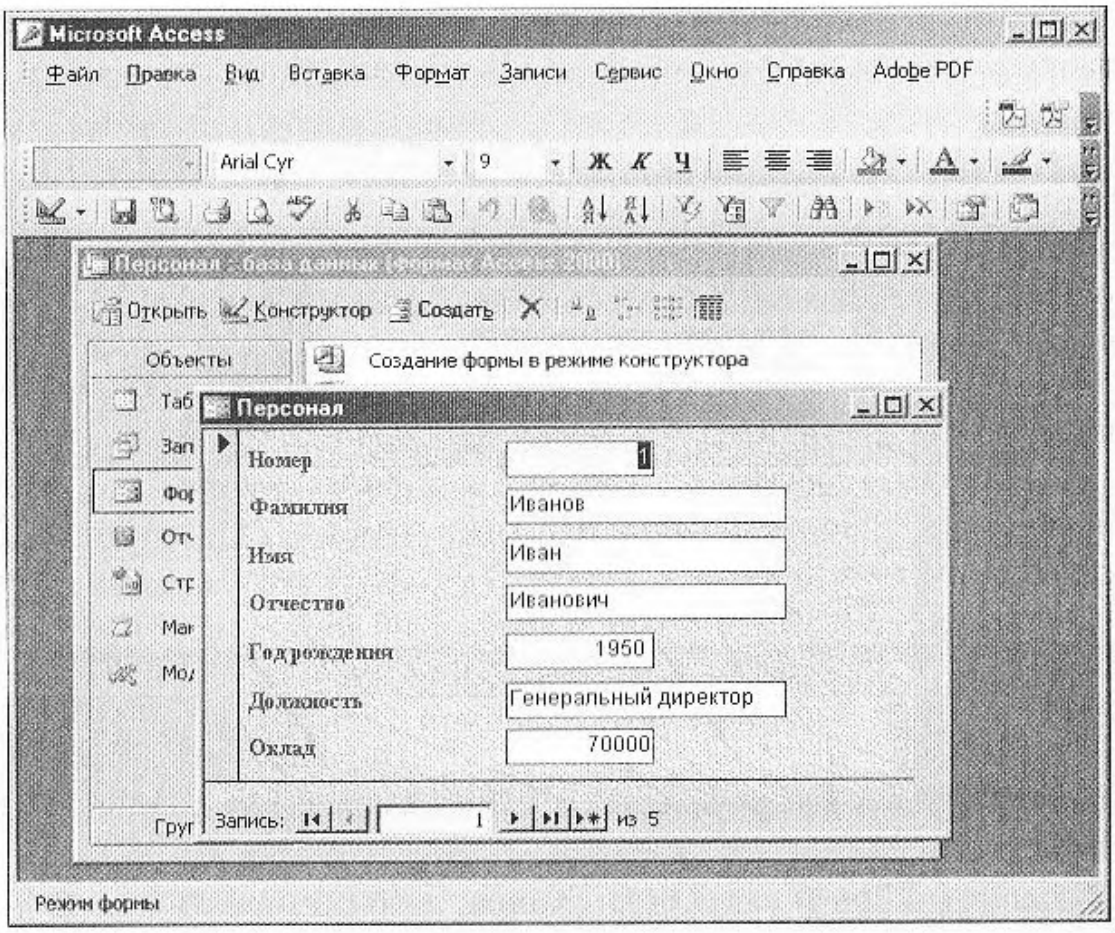


Рис. 5.32. Форма для базы данных «Персонал»

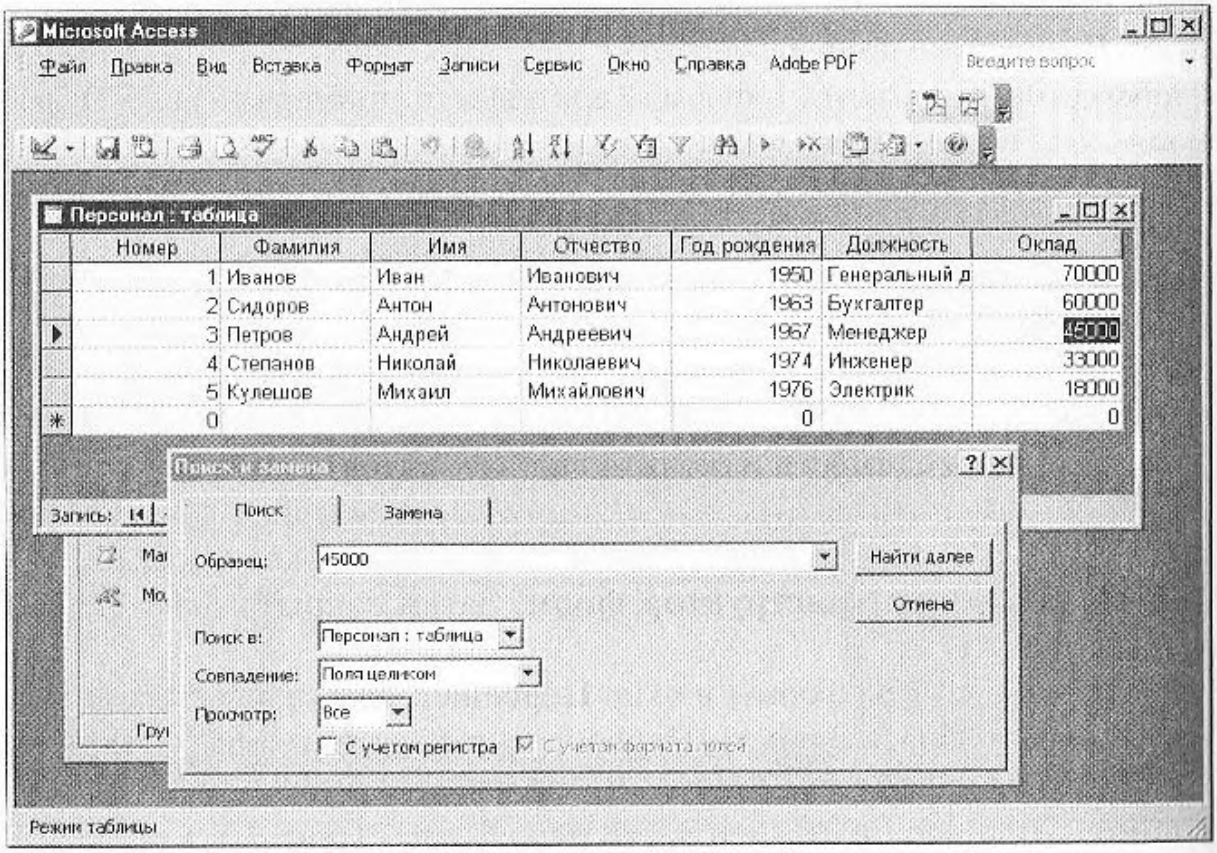


Рис. 5.33. Поиск по базе данных «Персонал»

1. Открыть таблицу «Персонал»: в окне **Персонал: база данных** на вкладке **Таблицы** выбрать таблицу «Персонал» и нажать кнопку **Открыть**.

2. Выбрать команду **Правка**, **Найти**. В поле **Образец** ввести искомую информацию (например, 45 000), в поле **Поиск в** выбрать значение **Персонал: таблица**, а в поле **Совпадение** — **Поля целиком**. Нажать кнопку **Найти далее**. При этом выделится то поле, значение которого совпадает с заданным (45 000) (рис. 5.33).

Можно осуществлять *поиск по условиям*. Условия формируются с помощью операций отношения =, <>, >=, <=, >, < и т.д. Такой поиск называют также *поиском по фильтрам*. Простые фильтры содержат одно условие поиска, сложные — несколько. Условия связываются логическими операциями.

Для быстрого поиска записей используют фильтр. Например, чтобы найти записи о работниках, имеющих оклад менее 45 000 руб. и родившихся после 1967 г., следует:

1. Открыть таблицу «Персонал»: в окне **Персонал: база данных** на вкладке **Таблицы** выбрать таблицу «Персонал» и нажать кнопку **Открыть**.

2. Выбрать команду **Записи**, **Фильтр**, **Изменить фильтр**. В появившемся окне задать условия поиска. Фильтр создан (рис. 5.34).

3. Выбрать команду **Фильтр**, **Применить фильтр**. В появившемся окне появятся записи, удовлетворяющие условиям поиска (рис. 5.35).

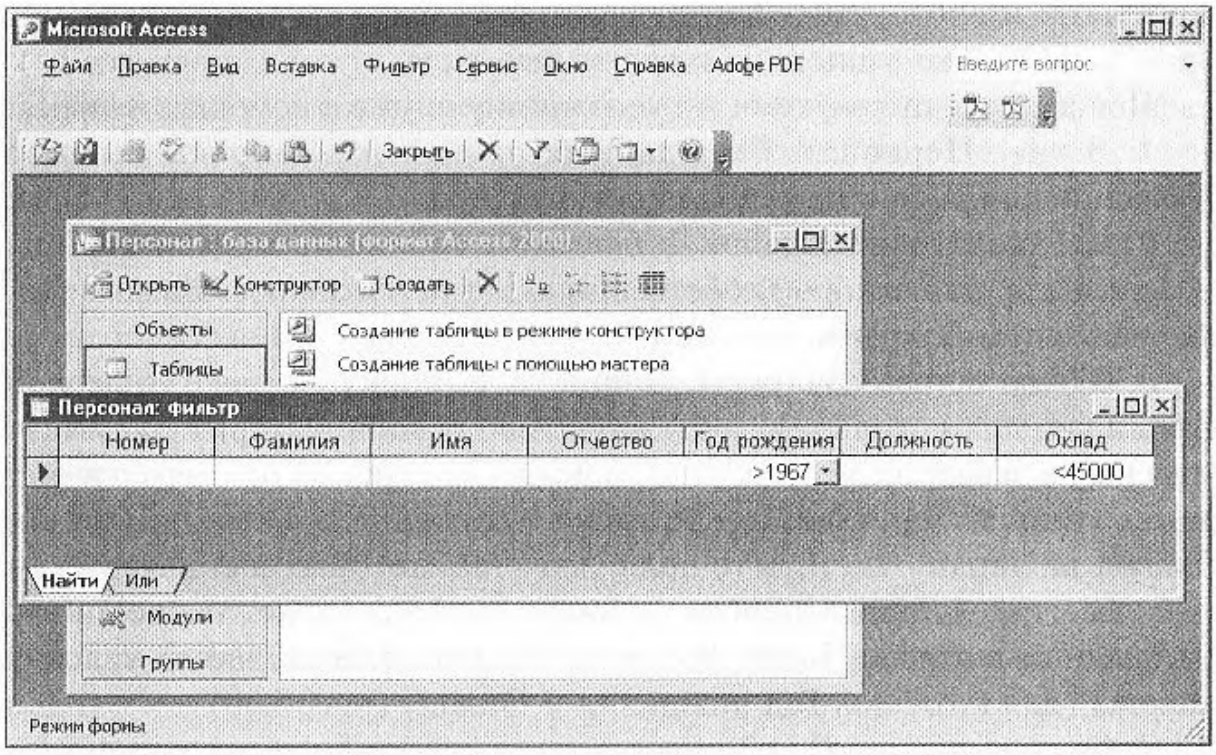


Рис. 5.34. Создание фильтра

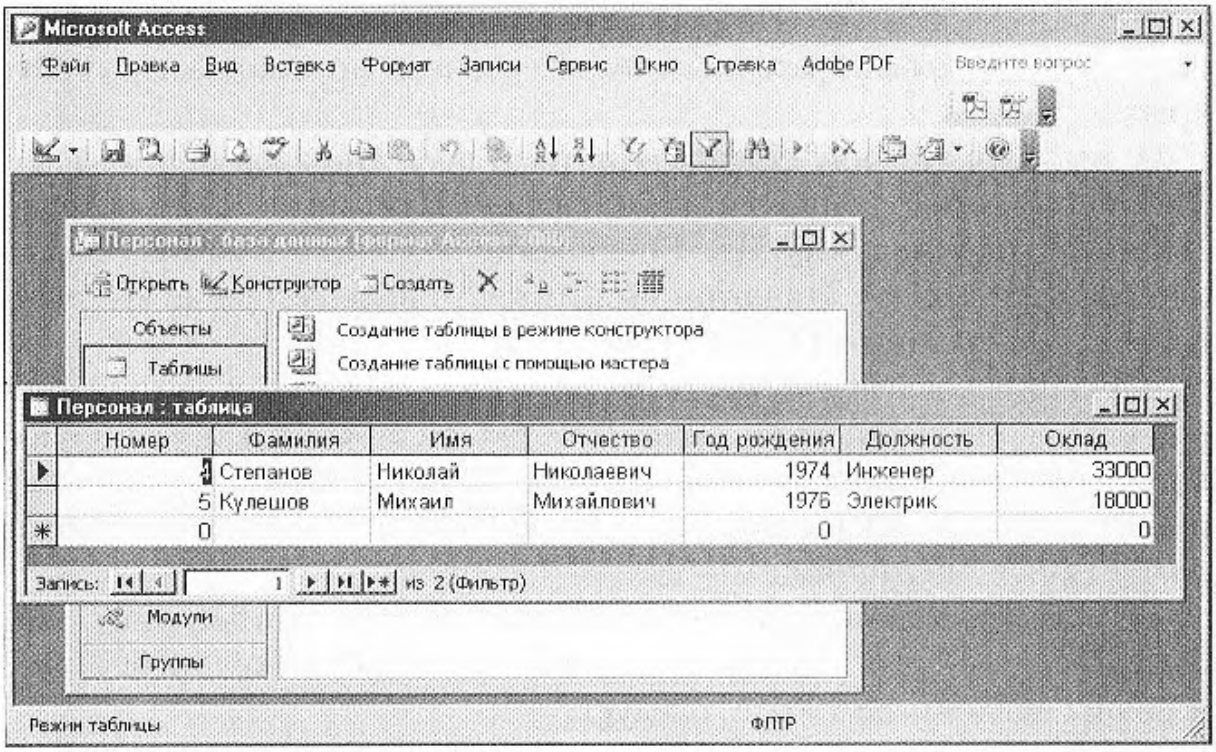


Рис. 5.35. Результат работы фильтра

Можно осуществлять *поиск с помощью запросов*. Практически это то же самое, что и фильтры, однако в отличие от последних они являются самостоятельными объектами БД, а фильтры «привязаны» к конкретной таблице. Результат работы запроса — таблица. Для запроса может быть создана форма. Так же как и фильтры, запросы могут быть простыми и сложными. В процессе выполнения запроса можно выбирать не только записи, но и отдельные поля.

Поиск проводится с теми же условиями, но уже с помощью запроса:

1. В окне **Персонал: база данных** на вкладке **Запросы** выбрать **Создание запроса с помощью конструктора**.

2. На диалоговой панели **Добавление таблицы** выбрать таблицу «Персонал», для которой собственно и создается запрос. Нажать **Добавить**, затем **Закрыть**.

3. В окне запроса **Запрос 1: запрос на выборку** в строке **Поле** указать поля, для которых задаются условия, и поля, которые будут выведены на экран после отработки запроса (для нашего примера следует выбрать поля: **Фамилия**, **Год рождения**, **Должность**, **Оклад**). В строке **Вывод на экран** отметить те поля, значения которых нас интересуют в качестве результата поиска (в данном случае отметить все поля, выбранные в строке **Поле**). В строке **Условия отбора** ввести условия поиска (для столбцов **Год рождения** и **Оклад**) (рис. 5.36).

4. Сохранить запрос под именем «запрос1», выбрав команду **Файл**, **Сохранить**. Закрыть окно **Запрос 1: запрос на выборку**.

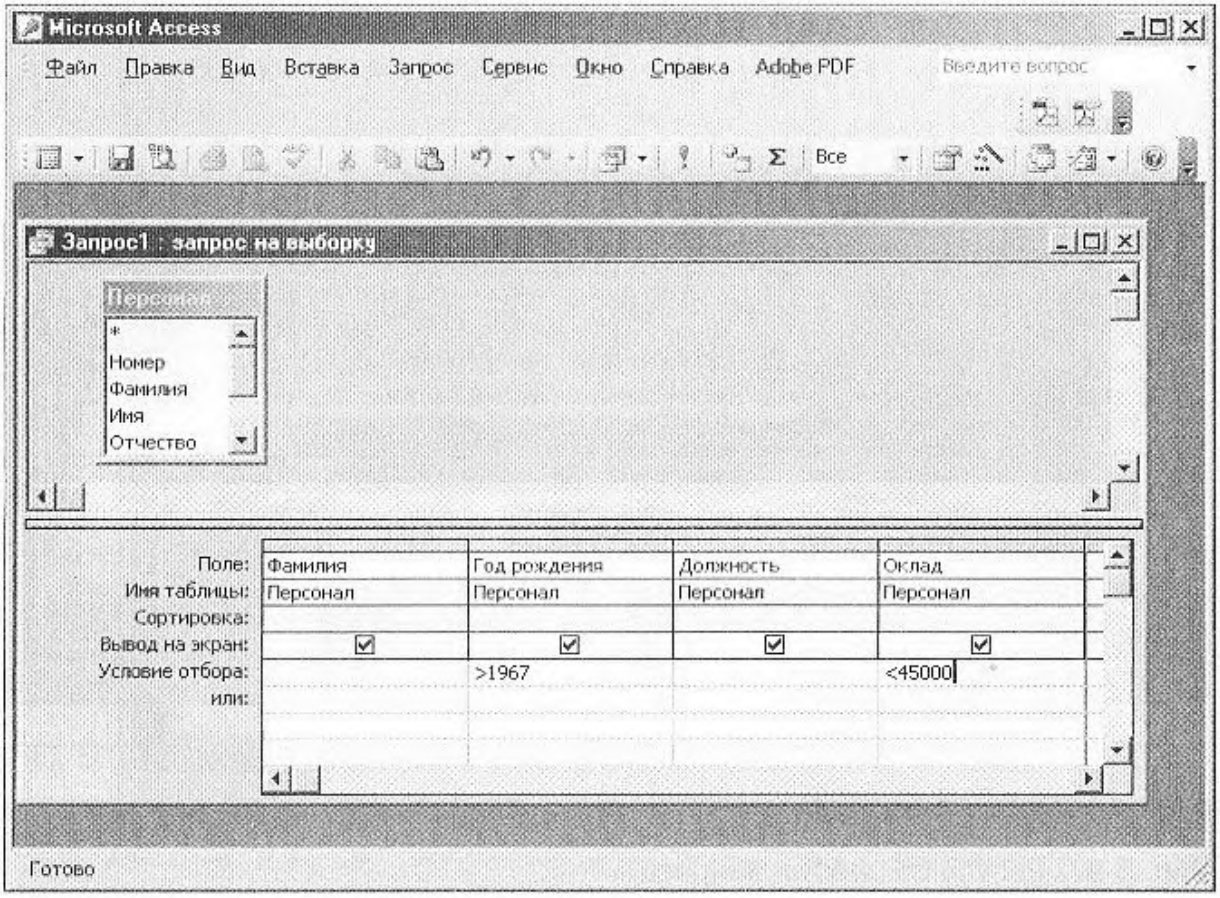


Рис. 5.36. Создание запроса на выборку

5. Для открытия окна запроса в окне **Персонал: база данных** на вкладке **Запросы** выбрать созданный «запрос1» и нажать **Открыть**. Появится окно с результатами выполнения запроса (рис. 5.37).

Базы данных могут содержать огромное число записей, и, естественно, периодически возникает потребность в их преобразовании к определенной последовательности. Такое преобразование называется *сортировкой*.

Обычно сортировка производится по конкретному полю. Значения, содержащиеся в этом поле для всех записей, могут быть рассортированы по алфавиту, значению или дате в зависимости от типа данных поля. Сортировка может производиться по возрастанию или по убыванию. При этом происходит соответствующее перемещение самой записи.

Для сортировки БД по возрасту сотрудников в порядке возрастания дат рождения необходимо:

1. Открыть таблицу «Персонал»: в окне **Персонал: база данных** на вкладке **Таблицы** выбрать таблицу «Персонал» и нажать кнопку **Открыть**.

2. Выделить в таблице «Персонал» столбец **Год рождения**, кликнув левой кнопкой мыши на названии столбца. Выбрать команду **Записи**,

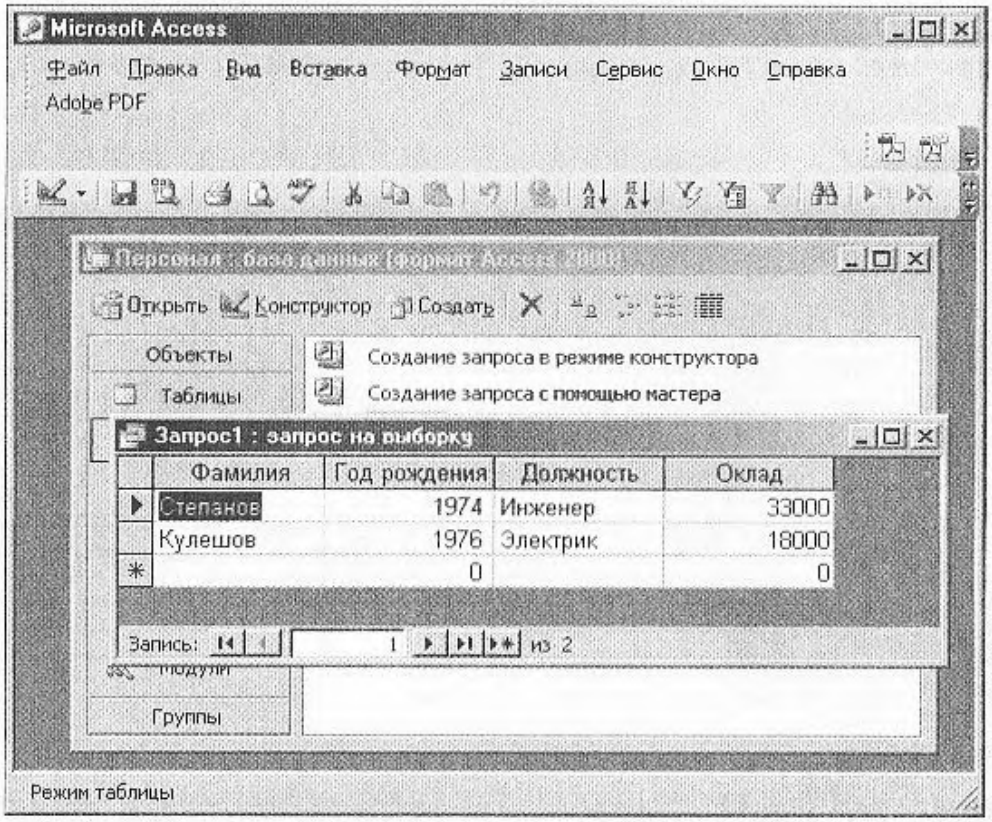


Рис. 5.37. Результат работы запроса

Сортировка, **Сортировка по убыванию**. Записи будут отсортированы по возрастанию значения содержимого этого поля (рис. 5.38).

Сортировку также можно производить с помощью специальных запросов.

Вывод информации из БД на принтер осуществляется непосредственно командами **Файл**, **Печать**. Однако для более качественного представления выходной информации используют *отчеты*. Отчеты — это производные объекты, которые создаются на основе таблиц, запросов и форм. Создают отчет по следующему алгоритму:

1. В окне **Персонал: база данных** на вкладке **Отчеты** выбрать **Создание отчета с помощью мастера**.

2. В открывшемся окне **Создание отчетов** выбрать в области **Таблицы и запросы** таблицу «Персонал». Из области **Доступные поля** в область **Выбранные поля** перенести те поля выбранной таблицы, которые необходимо отобразить в отчете (для рассматриваемого примера выбрать поля **Фамилия**, **Имя**, **Отчество**, **Должность**, **Оклад**). Щелкнуть **Далее**. В нашем случае на панелях добавления уровней группировки и задания порядка сортировки ничего не менять.

3. Перейти к панели выбора макета для отчета. Выбрать макет «табличный», ориентацию страницы — «книжная». Щелкнуть **Готово**. Появится окно созданного отчета «Персонал» (рис. 5.39).

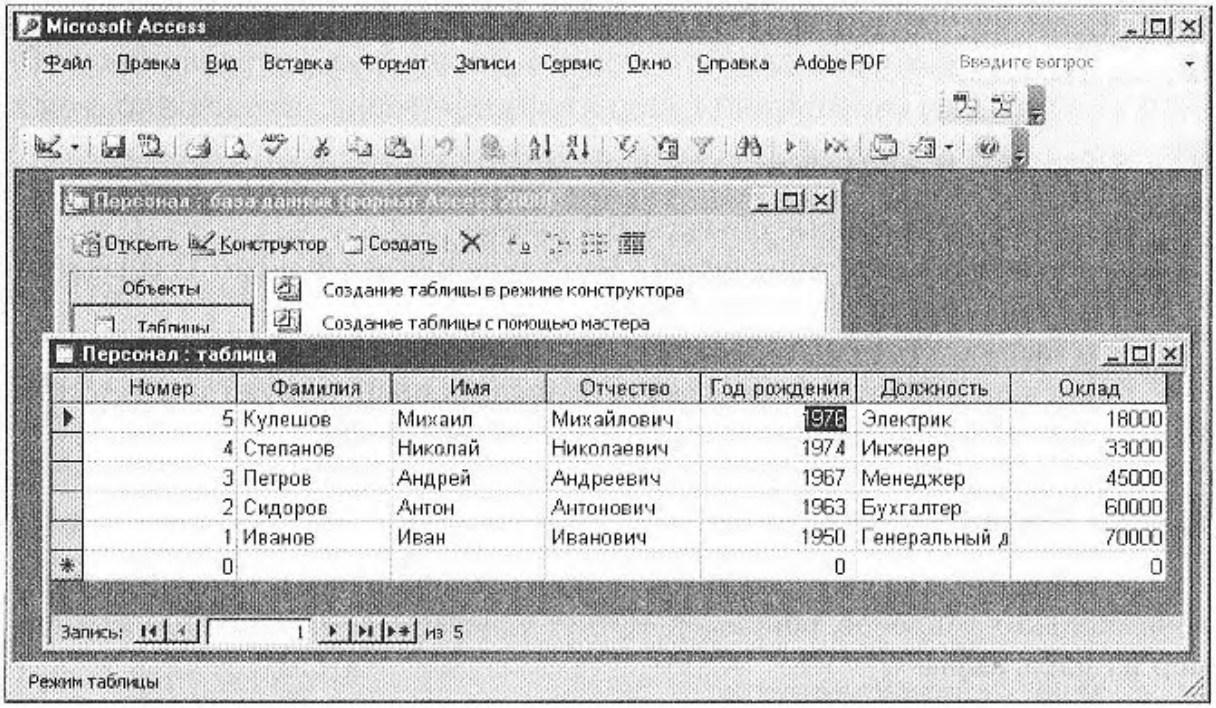


Рис. 5.38. Сортировка данных в таблице

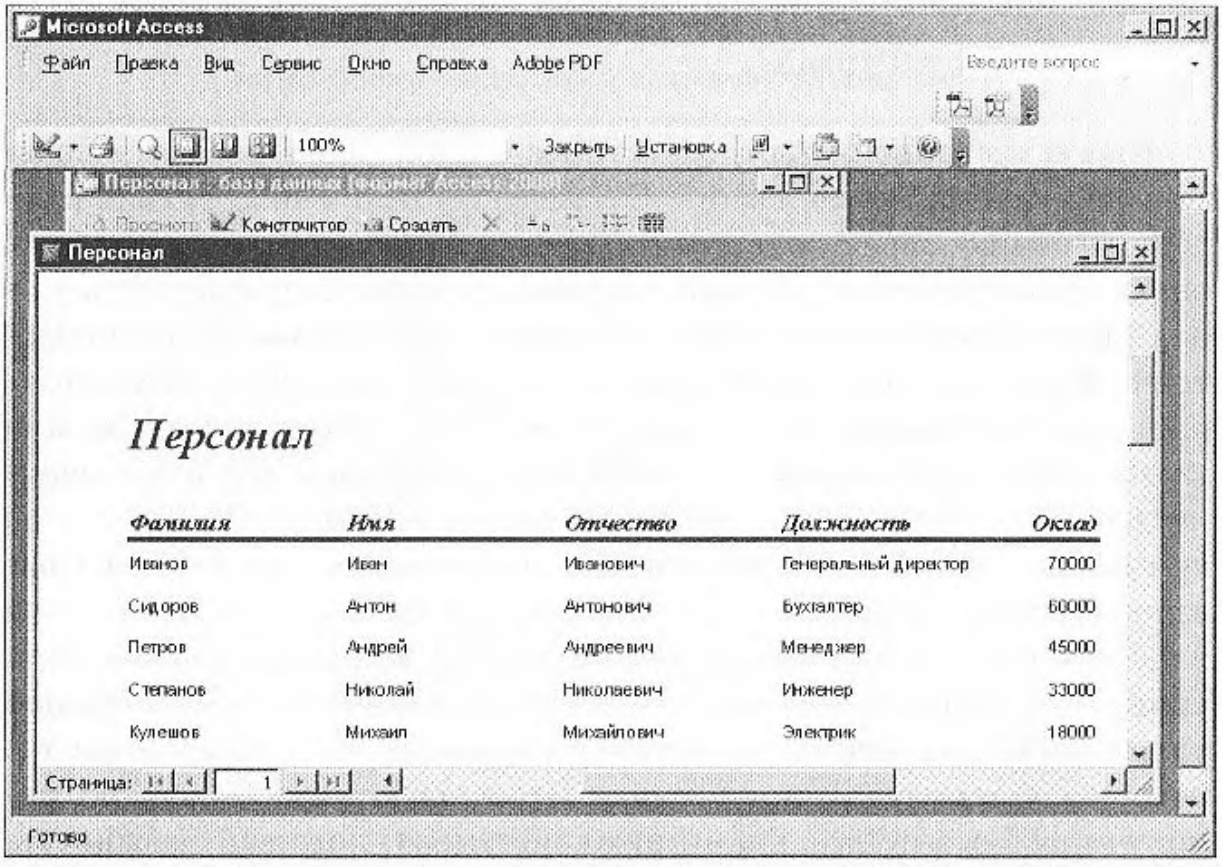


Рис. 5.39. Форма отчета по базе данных «Персонал»

4. Закрыть окно отчета «Персонал».

5. В окне **Персонал: база данных** на вкладке **Отчеты** выбрать отчет «Персонал» и нажать **Просмотр**. Отчет откроется в режиме предварительного просмотра.

6. Если вас устраивает вид отчета, для печати выбрать команду **Файл, Печать**.

? Вопросы и задания

1. Перечислите основные объекты БД Access.
2. Какие существуют типы полей? Что такое ключевое поле? Охарактеризуйте свойства каждого типа поля.
3. Что такое СУБД?
4. Какие инструменты для поиска информации присутствуют в БД Access?
5. Опишите поля адресной книги в мобильном телефоне.

П-СЭ Проектное задание

Создайте БД для своего курсового проекта, подготовьте отчеты на основе нескольких запросов, например «Отчет успеваемости». Используйте пример из параграфа.

5.6. Модель расчета оплаты труда в табличной базе данных

Задача расчета оплаты труда в той или иной форме решается практически в каждой организации. Ее смысл состоит в том, что для каждого работника определяются виды начислений, которыми могут быть оклад, фиксированные премии, разовые премии, возврат долга работнику и т.д. При этом к видам удержаний могут быть отнесены подоходный налог, возврат долга организации, выплаты по исполнительным листам, штрафы и др. Здесь и далее упростим систему расчета заработной платы, поскольку могут быть десятки видов начислений и удержаний. Кроме того, работодатель на основании начисленной суммы заработной платы перечисляет в налоговую службу страховые взносы и подоходный налог. Из этих средств производятся отчисления в Пенсионный фонд Российской Федерации и другие фонды и организации. Разница между общей суммой начислений и общей суммой удержаний конкретному работнику составляет сумму «К выдаче», которую он и получает под роспись. Чтобы работнику было ясно, как получилась эта сумма, формируют документ, который называется «Расчетно-платежная ведомость». Рассмотрим, как сформировать такой документ. Предположим, что необходимо сделать проект для Общества с ограниченной ответственностью (ООО) «Аврора», занимающегося реализацией и ремонтом компьютеров. Данный проект применимо к небольшой организации наиболее удобно реализовать с помощью электронных таблиц, так как информация в основном

имеет табличный характер и обрабатывается с помощью формул. Каждый работник организации имеет табельный номер — своего рода ключ, однозначно идентифицирующий конкретного работника и присутствующий во всех документах и расчетах, связанных с данным работником. Будем учитывать следующие виды начислений: должностной оклад, персональная премия от оклада, начисляемая в соответствии с установленным по организации процентом (величину этого процента вынесем в отдельную ячейку «Перс.пр.»), возврат долга за организацией работнику («Долг») и прочие начисления («Проч.нач»). К удержаниям отнесем: подоходный налог в размере 13 % начисленной суммы, штрафы, наложенные администрацией на данного работника, удержания по исполнительному листу, возврат долга работника организации, прочие удержания.

"Утверждаю"
Генеральный директор ООО "Аврора"
_____К.К.Птицын
"_____"_____2003 г

Установленный процент премии 26,30%

РАСЧЕТНО-ПЛАТЕЖНАЯ ВЕДОМОСТЬ ООО "АВРОРА" ЗА ОКТЯБРЬ 2003 Г

Nпп	Таб N	ФАМИЛИЯ И.,О.	ДОЛЖНОСТЬ	НАЧИСЛЕНИЯ					
				Оклад	Премия%	Перс.пр.	Долг	Проч.нач	Всего.начис
1	2	3	4	5	6	7	8	9	10
1	1001	Птицын К.К.	Генеральный дире	25000	6575	2000		1500	35075
2	1003	Воронин К.Л.	Технический дирек	19000	4997	2000			25997
3	1004	Сорокин Н.Д.	Начальник техотде	16000	4208	1000			21208
4	1006	Соловьев П.Р.	Инженер	12000	3156	1000	750		16906
5	1007	Дроздов И.В.	Инженер	12000	3156				15156
6	1010	Синицын Ю.А.	Инженер	11500	3024,5			2750	17274,5
7	1011	Петухов Е.Л.	Программист	12500	3287,5	2500			18287,5
8	1012	Кукушкина А.В.	Начальник торготд	14000	3682	1000			18682
9	1014	Воробьева Л.П.	Продавец-консуль	11000	2893				13893
10	1015	Соколов Н.А.	Продавец-консуль	11500	3024,5	500			15024,5
11	1017	Орлов Ф.П.	Продавец-консуль	12000	3156			1430	16586
12	1018	Дятлов О.Ф.	Заведующий склад	13000	3419		2500		18919
				169500,00	**44578,50**	**10000,00**	**3250,00**	**5680,00**	**233008,50**

	УДЕРЖАНИЯ							К выдаче	Подпись
	Аванс	Под.налог	штрафы	Удерж. по исп.лист	Долг	Проч.удерж	Всего удержано		
	11	12	13	14	15	16	17	18	19
1	10000	4559,75					14559,75	20515,25	
2	7600	3379,61		6499,25			17478,86	8518,14	
3	6400	2757,04				524,5	9681,54	11526,46	
4	4800	2197,78			334		7331,78	9574,22	
5	4800	1970,28					6770,28	8385,72	
6	4600	2245,69	500				7345,685	9928,815	
7	5000	2377,38	750				8127,375	10160,125	
8	5600	2428,66					8028,66	10653,34	
9	4400	1806,09			276		6482,09	7410,91	
10	4800	1953,19					6553,185	8471,315	
11	4800	2156,18				616	7572,18	9013,82	
12	5200	2459,47	500	1891,9			10051,37	8867,63	
	67800,00	**30291,11**	**1750,00**	**8391,15**	**610,00**	**1140,50**	**109982,76**	**123025,745**	

Рис. 5.40. Модель базы данных на примере расчетно-платежной ведомости

На рис. 5.40 представлена модель рассмотренной базы данных.

Практически каждая графа имеет итоговые суммы, которые необходимы для учета в бухгалтерской отчетности, заказа наличных денег в банке, перевода в налоговую инспекцию. Кроме того, работодатель должен перевести в налоговые органы налоги (предположим, что общий размер налоговых отчислений составит 34,8 % начисленной суммы).

В качестве наиболее известных специальных бухгалтерских программ можно упомянуть «Турбобухгалтер», «Инфобухгалтер», «1С: Бухгалтерия», «Бест», «Парус». Эти программы являются *автоматизированными системами управления* финансовой информацией. В таких БД заложены автоматические инструменты и все необходимые параметры финансовых расчетов, включая налоговые начисления, расчет премий, начислений за очередной отпуск, различные льготы и страховые начисления.

На примере АСУ «Парус» (демоверсия доступна по ссылке: http://www.oknemika.ru/body1100.asp) попробуем отследить основные этапы работы. При инсталляции программы на экране появляется окно регистрации пользователя.

Для работы с программой необходимо выполнить следующие действия:

1. Создать БД, в которой указать всю необходимую условно-постоянную информацию для работы бухгалтерской программы: данные сотрудников, организации-контрагенты, объекты бухгалтерского учета и т.д. (рис. 5.41).

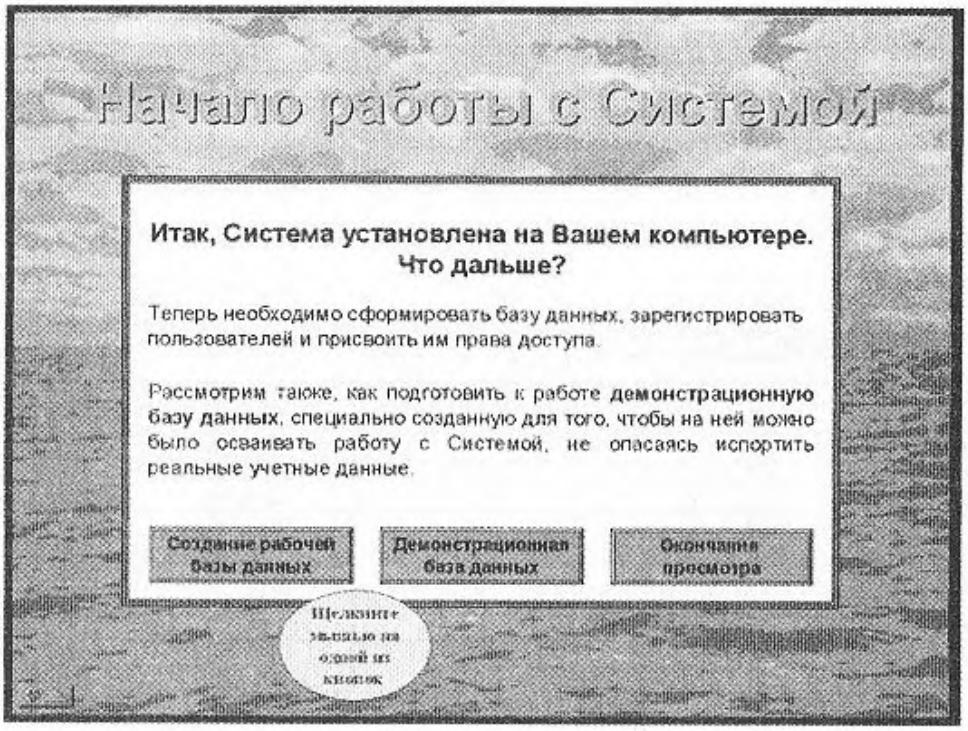


Рис. 5.41. Начало работы с системой «Парус»



Рис. 5.42. Задачи настройки



Рис. 5.43. Отработка процесса закупки товара

2. После того как БД создана, соответствующим образом настроить систему на специфику вашей организации (рис. 5.42).
3. Указать в системе типовой бизнес-процесс вашей организации (нажимая каждую кнопку, вы внесете в систему свои данные) (рис. 5.43).



Рис. 5.44. Типовая схема работы модуля «Зарплата»

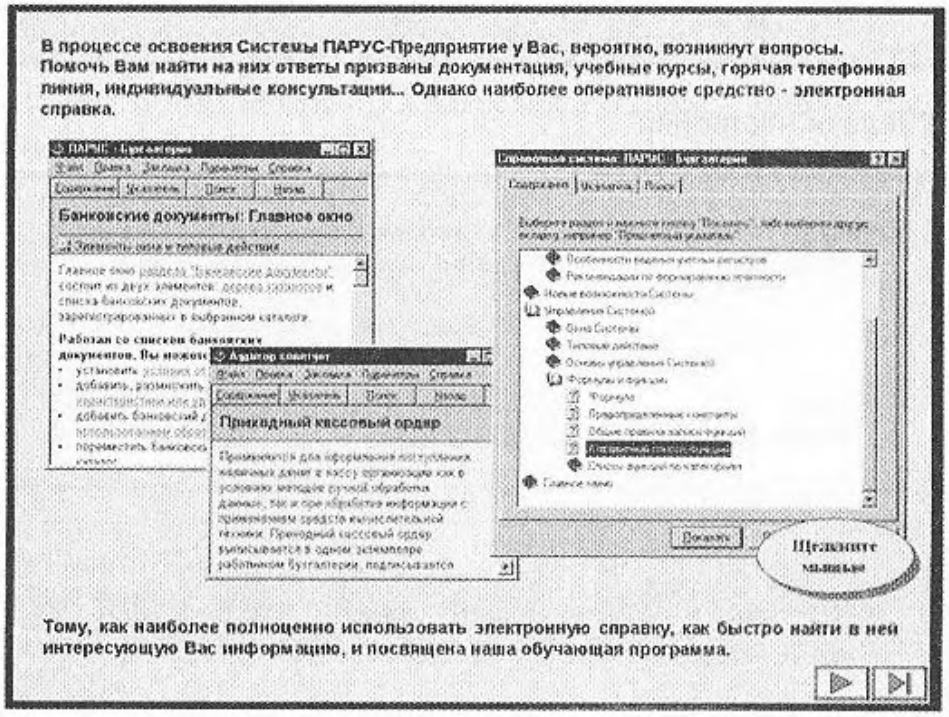


Рис. 5.45. Работа с электронной справкой

4. Для работы, например, модуля расчета заработной платы ввести данные, нажимая последовательно на кнопки **Формирование списка подразделений**, **Формирование списка сотрудников** и т. д. (рис. 5.44).

При работе с программой неизбежно возникновение различного рода вопросов, которые поможет разрешить электронная справка (рис. 5.45). Необходимо отметить, что каждая серьезная бухгалтерская программа имеет свой встроенный электронный учебник по ее использованию.

? Вопросы и задания

1. Поясните суть автоматизированной системы управления. Обоснуйте свой ответ на примере любой из известных программ. Используйте электронный учебник к ней на сайте поддержки.
2. Выполните задание на компьютере, используя рассмотренную нами модель «Расчет заработной платы».
3. Как с помощью системы запросов оформить отчеты: «Общая ведомость», «Личный расчетный листок»?

П-СЭ Проектное задание

Выполните проект «Мой интернет-магазин».

Постройте модель таблицы. Определите ассортимент товара вашего магазина. Заполните таблицу: наименование товара, стоимость единицы товара, количество единиц товара. Добавьте в таблицу столбец для пометок заказа покупателя (представьте, что заказ оформляется по сети) с учетом количества выбранного товара и вычислите стоимость покупки на основе расчета помеченных объектов — товаров и их количества.

Этап 1. «Прайс-лист». Сформируйте отчет: прайс-лист для магазина, учитывая курс валюты — параметр задачи, который меняется ежедневно.

Оформите отчет: чек с заставкой — маркой фирмы, по итогам оформления заказа покупателя.

Этап 2. «Товарная ведомость». Определите выручку за день с учетом количества единиц, купленных на данный день. Оформите отчет в форме таблицы с ассортиментом товара и отражением в ней количества единиц товара, купленных на данный день.

Этап 3. «Зарплата». Пусть в вашем магазине работают пять сотрудников: продавец, кассир, администратор, охранник, уборщица.

Постройте таблицу для расчета заработной платы и по ней сделайте отчет — ведомость на заработную плату. Отразите в отчете-ведомости ФИО. сотрудника, ставку сотрудника, количество рабочих дней в данном месяце, количество отработанных дней, налог, пенсионный взнос, к выдаче, подпись.

Этап 4. Распечатайте все отчеты.

ГЛАВА 6 ТЕЛЕКОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

6.1. Компьютерная сеть как средство массовой коммуникации

Явление развития компьютерных сетей как следствие информатизации общества интересно и требует особого внимания. В своем сознании мы тесно связываем понятие персонального компьютера с компьютерной сетью. Если стоящий дома компьютер по каким-либо причинам не подключен к Интернету, то можно говорить о его неэффективном использовании, так как теряется его важнейшая функция — коммуникативная, т. е. взаимодействие с людьми и информационными ресурсами общества.

Идея создания ПК впервые была воплощена в жизнь в середине 1970-х гг. Наряду с созданием вычислительных машин коллективного пользования с очень большими объемами оперативной памяти появилось тяготение к проектированию индивидуальных машин для управления технологическими процессами и обработки экспериментальных данных в исследовательских лабораториях. Были созданы малые вычислительные машины, так называемые мини-ЭВМ, но применялись они только в научной сфере. Мини-ЭВМ, соединенные линиями связи с мощными вычислительными комплексами коллективного пользования, могли использоваться как терминалы — точки входа в сетевое взаимодействие. Это был один из первых шагов к формированию компьютерных сетей.

Основная цель создания компьютерных сетей заключается в обеспечении обмена данными между вычислительными машинами, входящими в сеть, поэтому первоначально подобные сети в США связывали именно научные центры — университеты.

Первая сеть с коммутацией пакетов (процесс соединения абонентов сети с передачей данных) была разработана в Англии в 1968 г. в Национальной физической лаборатории. Первая многоузловая сеть

с коммутацией пакетов Arpanet вступила в действие в США в 1969 г. Эксперимент с Arpanet был настолько успешен, что многие организации захотели войти в нее в целях использования для ежедневной передачи данных. И в 1975 г. Arpanet превратилась из экспериментальной сети в рабочую (рис. 6.1).

В 1971 г. была создана сеть Alocha (Гавайи, США), в которой реализовались методы передачи пакетов по радиоканалам. Модель сети Ethernet разработали сотрудники фирмы *Xerox* в 1974—1976 гг. Протокол этой сети был стандартизирован в 1980-х гг. В течение 1974—1982 гг. ряд ведущих компьютерных фирм США разработал архитектуры и сетевые технологии, повлиявшие на формирование современных сетей. Фирма *DEC* в 1975 г. создала сеть Decnet, развивавшуюся вплоть до 1990 г. В 1982—1988 гг. университеты и фирмы США создали сеть Bitnet, получившую всемирное распространение.

Итак, историю Интернета можно разделить на несколько этапов.

1961—1970 гг. Разработаны технические принципы компьютерной сети, введен в действие Arpanet.

1971—1980 гг. Придуман знак @. Написана первая программа для электронной почты. Осуществлена первая международная связь по электронной почте между Англией и Норвегией. Число узлов Arpanet возросло до нескольких десятков, проложены специальные кабельные линии, соединяющие некоторые узлы, начинает функционировать электронная почта, о результатах работ ученые докладывают на международных научных конференциях.

1981—1990 гг. Принят протокол TCP/ IP, введена система доменных имен Domain Name System (DNS). Сформирована сеть Интернет.

1991—2000 гг. Ethernet стал самой распространенной технологией локальной компьютерной сети. Интернет объединил локальные сети и стал средством массовой коммуникации. Произошло сращивание сотовой связи и Интернета. Телетехнологии встроены в глобальную сеть — телемосты, видеоконференции. Услуга Интернет и электронной почты встроена в мобильные телефоны.

2001—2010 гг. Произведено массовое подключение отдельных пользователей и локальных сетей к Интернету,

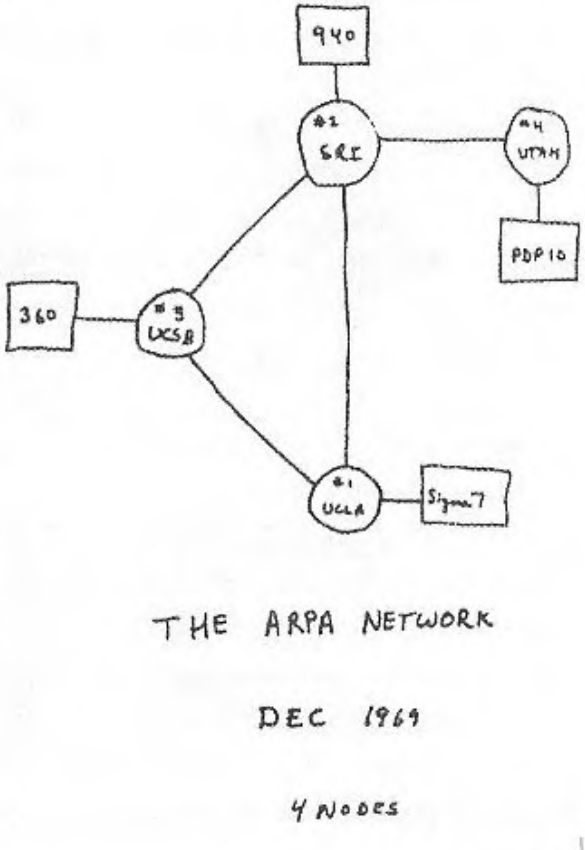


Рис. 6.1. Документальный эскиз сети Arpanet, состоящей из четырех узлов. 1969 г.

в том числе всех образовательных учреждений. Произошло повсеместное подключение домовладений к Интернету. Осуществлено развитие государственных программ «Цифровое государство» и «Электронное правительство». Разработана юридическая база для электронной подписи. Созданы и массово распространены сетевые социальные сервисы в сфере СМИ, цифрового телевидения, банковских, страховых услуг, услуг интернет-телефонии, интернет-магазинов, цифровых коллекций видео, цифровых библиотек, дистанционного образования, социальных сетей.

Если ранее компьютерная сеть использовалась исключительно в качестве среды передачи файлов и сообщений электронной почты, то сегодня решаются более сложные задачи *распределенного доступа к ресурсам*. Этот доступ предполагает выделение *сервера* (от англ. *to serve* — служить, обслуживать) — мощного обслуживающего сеть компьютера как узла управления доступом к информационным массивам компьютерной сети и компьютерам клиентов компьютерной сети, пользователей информационных массивов.

Технология распределенного доступа иначе называется *технологией клиент-сервер* (рис. 6.2). Она известна уже довольно длительное время, но раньше чаще всего использовалась в *локальных компьютерных сетях*, объединяющих компьютеры конкретного предприятия, учреждения, жилого дома и т. п. Клиенты, решившие подключить свой компьютер к компьютерной сети, получают права доступа: имя — *логин* и личный ключ входа в сеть — *пароль*. Установленный пароль пользователь не должен сообщать другим лицам. Технология

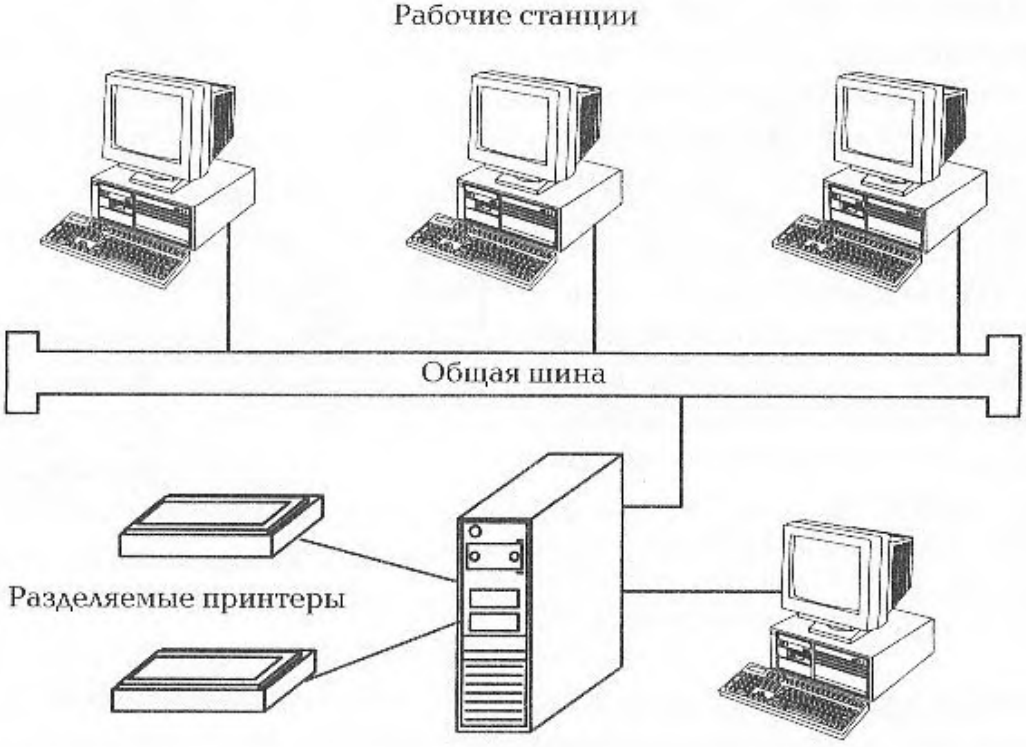


Рис. 6.2. Локальная компьютерная сеть архитектуры «клиент-сервер»

«клиент-сервер», с одной стороны, позволяет обеспечить защиту информации на сервере от несанкционированного использования и повреждения, а с другой — ограждает проникновение пользователей к личной информации клиента, не выложенной клиентом для общего пользования. Строгие требования к защите информации связаны с тем, что подключенный к сети ПК становится доступным из любой точки сети, и поэтому несравнимо более подвержен поражению вирусами и несанкционированному доступу. Так, несоблюдение режима защиты от несанкционированного доступа может привести к утечке информации, а несоблюдение режима защиты от вирусов — к выходу из строя важных систем и уничтожению результатов многодневной работы.

Сегодня, с развитием *глобальной компьютерной сети*, технология «клиент-сервер» уже рассматривает глобальную сеть как объединение различных локальных компьютерных сетей. Так, Интернет можно представить себе в виде мозаики, сложенной из небольших сетей разной величины, которые активно взаимодействуют одна с другой, пересылая файлы, сообщения и т. п. Это оказывается очень удобным для фирм, имеющих свои филиалы по всему миру, транснациональных корпораций и структур управления, а также простых пользователей из разных уголков планеты. В глобальной сети серверы локальных сетей также являются клиентами более высокого уровня. Для управления распределенным доступом клиентов низкого уровня и клиентов высокого уровня друг к другу в глобальной сети Интернет используются специальные устройства — *маршрутизаторы*. Любой сигнал и данные (пакеты) от одного компьютера сети к другому проходят через это устройство, подключенное к компьютеру. Маршрутизатор определяет самый оптимальный путь прохождения сигнала от компьютера к компьютеру. В настоящее время используются практически все известные линии связи (телекоммуникационные сети): от низкоскоростных телефонных линий до высокоскоростных цифровых спутниковых каналов, через которые идет сигнал от маршрутизатора.

Большинство телефонных линий в период развития Интернета имели аналоговый характер, но компьютерная техника является цифровой, т. е. для использования таких телефонных линий нужны были устройства, осуществляющие преобразование электрических сигналов из аналоговой формы в цифровую и обратное преобразование. Преобразование аналогового сигнала в цифровой осуществляется *модулированием* (чаще всего амплитудным), т. е. через определенные интервалы времени фиксируются значения амплитуды сигнала (как отмечалось ранее, они называются дискретами). Это уже цифровой

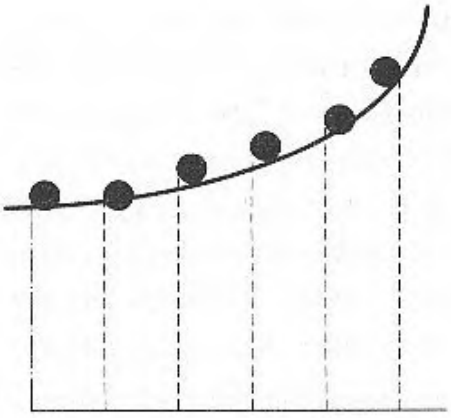


Рис. 6.3. Преобразование дискретного сигнала в непрерывный

сигнал. Обратное преобразование дискретов в непрерывный сигнал происходит аналогично, путем формирования линии, огибающей вершины дискретов (рис. 6.3). Такой процесс осуществляет демодуляцию. Именно этот механизм и определил название устройства-преобразователя *модем* (**мо**дулятор-**дем**одулятор).

Операционные системы, используемые в Интернете, также отличаются разнообразием, но все они в итоге обеспечивают цепочку взаимодействия: компьютер — модем — маршрутизатор — линия связи — маршрутизатор — модем — компьютер. Клиентом Всемироной сети может быть как отдельный пользовательский компьютер, так и сервер локальной компьютерной сети, внутри которой осуществляется посылка сигнала конкретному пользователю — клиенту нижнего уровня. Напомним, что такая модель управления сигналами называется *управлением с обратной связью*. Именно такую модель и реализовала компьютерная сеть.

Работа вычислительных сетей, т.е. обмен данными и взаимосвязь клиентов, выполняется в соответствии с достаточно сложными *протоколами* (правилами) взаимодействия. Протоколы необходимы при разработке и управлении сетью и позволяют осуществлять соединение и обмен данными между двумя и более включенными в сеть устройствами.

При разработке Интернета был создан *межсетевой протокол TCP/IP* (Internet Protocol), который обеспечивает подключение к адресату и надежную передачу данных именно по указанному адресу между различными подсетями с выбором кратчайшего пути передачи. Этот протокол также поддерживает центральную справочную службу, передачу экстренной и управляющей информации. *Транспортный протокол TCP* (Transmission Control Protocol) — это протокол управления передачей, который осуществляет передачу пакетов (потоков данных) от компьютера к компьютеру через Интернет. TCP работает на верхнем уровне между двумя конечными системами, например, интернет-пользователь и интернет-сервер.

В глобальной сети Интернет существует большое число сервисов. WWW — самый популярный сервис Интернета и удобный способ работы с информацией через web-сайты. Пользуясь несложным языком описания, можно составлять гипермедийные документы для их последующей публикации в Интернете (под гипермедийным подразуме-

вается электронный документ, который может содержать все виды информации — от простого текста до мультимедийных роликов) на основе *протокола HTTP* (HyperText Transfer Protocol — протокол передачи гипертекста). Для того чтобы увидеть содержание документа, нужно иметь на компьютере-клиенте программу просмотра — *браузер*, поддерживающую многие *расширения HTML* (Hyper Text Markup Language — язык гипертекстовой разметки документов). Именно с помощью этого языка представления информации оформляется информация для *web-сайтов* (WWW).

Тим Бернерс-Ли написал первый браузер (который он и назвал World Wide Web) и первый web-сервер (info.cern.ch). В 1991 г. это ПО стало доступно достаточно широкому кругу пользователей Интернета. Он также разработал систему адресации к web-документам в Интернете. Тим Бернерс-Ли назвал ее URI (Universal Resource Identifier — универсальный идентификатор ресурсов). Сейчас ее называют *URL* (Uniform Resource Locator — унифицированный указатель ресурса).

Важное место в системе адресации занимает символьное имя, которое может иметь сложную иерархическую структуру, например http://metodist.lbz.ru. Крайний справа элемент — *имя домена* верхнего уровня, которое известно во всей глобальной сети Интернет. Имя домена верхнего уровня определяется по территориальному принципу (ru — Россия, su — бывший СССР, usa — США, uk — Англия и т.п.) или организационному (com — коммерческая организация, org — некоммерческая, edu — образовательная, gov — государственная и т.п.). Например, информационная коллекция http://ru.wikipedia.org/ содержит детальную информацию о всех аспектах информационных технологий и компьютерных сетей.

Таким образом, разработка языка HTML, протокола HTTP, web-сервера, браузера и системы адресации URI явилась основой развития Всемирной паутины.

Появилась возможность организовать доступ многочисленных пользователей (клиентов) к HTML-документам, находящимся на серверах глобальной сети, которые создают различные компании, обслуживающие информационные потоки, — СМИ, интернет-магазины, авиакассы, правовые БД, библиотечные фонды, музейные коллекции и многие другие. Ясно, что и на компьютере пользователя, с которого происходит обращение к серверу, и на сервере должно быть установлено специальное ПО: сервер должен не только хранить документы, но и уметь быстро находить их по запросу клиента, клиент (браузер) должен распознать HTML-код и визуализировать его, представить в легко воспринимаемом виде.

Вопросы и задания

1. В чем заключается основная цель создания компьютерных сетей?
2. Перечислите основные факторы, повлиявшие на возникновение интегрированных вычислительных сетей.
3. Какие существуют виды компьютерных сетей? Охарактеризуйте их.
4. Что такое сетевой протокол и каково его назначение? Что такое протокол TCP/IP?
5. Каков механизм взаимодействия компьютеров в сети? Каково назначение браузера? Что такое доменное имя?

6.2. Локальная вычислительная сеть

Вы узнали о двух различных подходах к созданию компьютерных сетей. *Глобальная сеть* строится таким образом, что процессы передачи данных, процедуры управления и административные службы отдельных подсетей не изменяются существенно. Каждая из подсетей сохраняет свою автономность, хотя требования к сетевому управлению и контролю ожесточаются. *Локальная сеть* проектируется как единая распределительная система, в которой приоритет отдается требованиям стандартности протоколов и эффективности общесетевых процедур управления.

При первом подходе объединение подсетей концентрируется в шлюзовых (межсетевых) устройствах. Сеть становится единой прежде всего с точки зрения пользователя. Такому подходу более всего соответствует глобальная сеть Интернет, реализованная механизмом виртуальных сетевых адресов компьютеров-клиентов.

Второй подход реализуется на практике в тех случаях, когда сеть физически соединяет компьютеры пользователей между собой и управляет ими на основе сервера. Совокупность компьютеров, обладающих возможностью информационного общения друг с другом, принято называть *локальной компьютерной* (*вычислительной*) *сетью* или *ЛВС* (LAN — Local Area Network). Самой распространенной технологией локальной компьютерной сети с середины 1990-х гг. стал Ethernet. Рассмотрим общую структуру ЛВС.

Любая ЛВС состоит из следующих аппаратных элементов:

- *компьютеры*, между которыми осуществляется связь, и внешние по отношению к ПК устройства, которые могут также входить в состав ЛВС (сетевые принтеры, сканеры, дополнительные «жесткие» диски, серверы различного назначения и т. п.);
- *кабели*, с помощью которых происходит соединение всех устройств в единую сеть. Это могут быть волоконно-оптические кабели, ка-

бели типа «витая пара», коаксиальные кабели или бескабельные системы передачи информации от одного устройства к другому (инфракрасные источники, Wi-Fi и др.);

- *узловые устройства*, позволяющие локальной сети нормально функционировать (коммутаторы, маршрутизаторы и т. п.).

Локальные вычислительные сети бывают одно- и двухранговыми (иерархическими).

Одноранговые сети удобно использовать в небольших организациях, в которых имеется несколько компьютеров. В этом случае все компьютеры могут быть равноправными по своим сетевым функциям и все пользователи могут получать свободный доступ к общим ресурсам других компьютеров. В крупных сетях с большим числом компьютеров организовывают *двухранговую* (*иерархическую*) *локальную сеть* с выделенным более мощным компьютером, который называется *сервером*. Таких серверов может быть несколько, и выполняют они разные функции. Все остальные компьютеры локальной сети в таком случае являются клиентами или рабочими станциями. При этом клиентский компьютер должен предоставлять пользователю возможность сформулировать запрос на сервер или сетевой ресурс, отправить запрос серверу, получить ответ, осуществить его интерпретацию и представить пользователю в понятной форме.

Сервер должен быть настроен таким образом, чтобы иметь возможность получить от клиентского компьютера запросы на сетевые ресурсы, выяснить полномочия клиента на выполнение того или иного запроса и, если клиент имеет полномочия, выполнить его запрос и передать ему результат выполнения этого запроса. Для решения всех этих задач и на клиентском компьютере, и на сервере должна быть установлена *сетевая операционная система*. При этом на клиенте и сервере могут быть установлены разные сетевые ОС, обеспечивающие управление каталогами и файлами, управление ресурсами сервера и всей сети в зоне ответственности данного сервера, коммуникативность, защиту от несанкционированного доступа как внутри сети, так и извне, отказоустойчивость составных частей ЛВС. В настоящее время наибольшее распространение получили сетевые ОС двух семейств: Unix и Windows.

Рассмотрим более детально составные элементы локальной сети.

Рабочая станция — это ПК, включенный в локальную сеть для осуществления обмена информацией. Он должен иметь сетевую плату, встроенный модем или устройство для приема сигнала без проводов (Wi-Fi-устройство, инфракрасный приемник и т. п.). Кроме того, на ПК должно быть установлено специальное ПО, позволяющее настроить его на прием и передачу информации в локальной сети.

Сервер — это специально выделенный компьютер для обработки запросов, поступающих от рабочих станций, который предоставляет рабочим станциям (клиентам) свои ресурсы (например, «жесткие» диски, прикладное ПО). На серверах устанавливается большой объем оперативной памяти, несколько дисков большого размера, а также высокопроизводительные процессоры. Рассмотрим некоторые специализации серверов.

Файловый сервер имеет диски большой емкости, к которым могут иметь доступ все компьютеры в сети. В этом случае информация хранится централизованно, причем с высокой степенью надежности, информация доступна с любого компьютера, подключенного к серверу, защищена от несанкционированного доступа, поскольку для подключения к серверу требуется пароль.

Сервер прикладных программ используется для выполнения прикладных программ пользователей. В этом случае нет необходимости приобретать и устанавливать пользовательские программы на каждый компьютер.

Сервер базы данных хранит и позволяет обрабатывать большие БД.

Сервер удаленного доступа дает возможность любому компьютеру, находящемуся далеко от офиса, работать так, как будто он находится в офисе.

Сервер печати служит для распечатывания документов на одном или нескольких общих (сетевых) принтерах. В этом случае отпадает необходимость подключать принтер к каждому компьютеру.

Сервер резервного копирования обеспечивает создание, хранение и восстановление копий данных, расположенных на файловом сервере и рабочих станциях.

Кроме того, существуют коммуникационный сервер, факс-сервер, почтовый сервер, которые могут находиться на одном или нескольких выделенных компьютерах.

Кабели предназначены для передачи информации по проводным каналам связи. Используются разные типы электрических проводников: коаксиальный кабель и витая пара, а также волоконно-оптический кабель. *Коаксиальный кабель* представляет собой проводник, заключенный в экранирующую оплетку. *Витая пара* представляет собой восемь проводников, свитых попарно и заключенных в общую изолирующую трубку. Такой кабель называется неэкранированной витой парой UTP (Unshielded Twisted Pair — неэкранированные скрученные пары). Выпускается и применяется также кабель, называемый экранированной витой парой STP (Shielded Twisted Pairs — экранированные скрученные пары). Применение экранов позволяет снизить уровень

помех, создаваемых таким кабелем, и увеличить помехозащищенность такого кабеля от внешних помех.

Передача информации по *волоконно-оптическим кабелям* наиболее применима в настоящее время. По своей конструкции волоконно-оптический кабель похож на коаксиальный, только вместо центрального металлического проводника располагается стеклянное или пластиковое волокно, по которому распространяется пучок света. В зависимости от траектории распространения луча света по оптическому волокну различают одномодовый (по кабелю идет один луч) и многомодовый волоконно-оптический кабель. В этом случае по кабелю проходит много лучей, движущихся по соседним траекториям.

При подключении компьютеров к ЛВС могут применяться **устройства беспроводной связи**. В этом случае отпадает необходимость прокладывать кабельные сети. Для подключения компьютеров используются сетевые адаптеры ввода-вывода с интерфейсами PCI (Peripheral Component Interconnect — взаимосвязь периферийных компонентов) и USB (Universal Serial Bus — универсальная последовательная шина). Беспроводной доступ может быть организован и с помощью обычных сетевых адаптеров, но при этом сетевой адаптер должен быть подключен к радиоприемнику-передатчику (точке беспроводного доступа) посредством стандартного соединительного шнура. Кроме того, точка доступа (хот-спот) играет роль центра беспроводной сети при объединении более двух компьютеров в беспроводную сеть. В таком случае она подключается к серверу.

В зависимости от частоты, на которой работает аппаратура беспроводной передачи, типа и мощности самой аппаратуры дальность устойчивой связи может составлять от нескольких десятков метров до нескольких десятков километров. Одной из тенденций развития внутрикорпоративных сетей является *беспроводная технология Wi-Fi* (Wireless Fidelity — беспроводная точность). В этом случае любая точка доступа включает в себя Wi-Fi-антенну для передачи радиосигнала, специальное Wi-Fi-оборудование — сетевой адаптер (беспроводная сетевая плата), беспроводные точки доступа (маршрутизаторы) и различные коммутационные устройства.

Bluetooth также является технологией беспроводных сетей. Физически типичное Bluetooth-устройство представляет собой радиоприемник и радиопередатчик, работающие на определенных частотах. Расстояние, на которое может быть установлено Bluetooth-соединение, невелико и составляет от 10 до 30 м. В настоящее время ведутся работы над увеличением этого расстояния. Преимущество Bluetooth состоит том, что для него не требуется прямой видимости или какой-либо направленной антенны, соединение может быть установлено даже

через стену. Главной же особенностью Bluetooth является то, что различные Bluetooth-устройства соединяются друг с другом автоматически. Для этого важно, чтобы Bluetooth-устройства находились достаточно близко друг к другу, обо всем остальном должны позаботиться сами Bluetooth-устройства и ПО. В настоящее время для включения в компьютерную сеть устройства с Bluetooth достаточно, чтобы Bluetooth был активирован. Примером работы с Bluetooth может служить использование мобильного телефона как сетевого клиента для пересылки информации из его памяти на компьютер, и наоборот, например фотографий, мелодий и пр.

Каждый компьютер, подключаемый к сети, должен быть оснащен *сетевым адаптером* (или сетевой платой). Самыми известными являются адаптеры следующих типов: Arc-Net, Token Ring, Ethernet. Из них последний используется в России наиболее широко.

Внутренний сетевой адаптер, представляющий собой плату, вставляется в разъем на системной плате компьютера. Кроме того, в настоящее время имеются сетевые адаптеры, подключаемые к USB-порту. Подключение такого адаптера к ПК возможно без отключения питания. Далее к сетевому адаптеру подключается линия связи ЛВС (например, витая пара). Для объединения компьютеров в сеть, состоящую более чем из двух компьютеров по физической топологии «звезда», необходимо устройство, исполняющее роль центра «звезды». Таким устройством чаще всего является *концентратор*, или *хаб* (от англ. *hub* — ступица).

Маршрутизатор (router) представляет собой устройство для разделения или объединения нескольких компьютерных сетей. Задача маршрутизатора — отфильтровывать информационные пакеты и пропускать с одного интерфейсного входа на другой только те пакеты, которые адресованы компьютером, находящимся в одной сети, компьютеру, находящемуся в другой сети.

Способ организации связи компьютеров ЛВС между собой называется **топологией локальной сети** (архитектурой, конфигурацией). Существует три основные топологии, входящие в состав ЛВС: «шина», «кольцо» и «звезда».

Топология «шина» предполагает подключение всех компьютеров к одному общему проводнику. На обоих концах такого проводника размещаются специальные согласующие устройства, называемые терминаторами. Основные преимущества данной топологии — дешевизна и простота монтажа. К недостаткам относятся сложность локализации места неисправности и низкая надежность: повреждение кабеля в любом месте приводит к прекращению обмена информацией между всеми компьютерами, входящими в сеть.

В *топологии «кольцо»* каждый компьютер сети связан с двумя близлежащими ПК. Достоинства и недостатки такой топологии такие же, как и для топологии «шина».

Топология «звезда» предусматривает прокладку для каждого компьютера в сети отдельного кабеля, соединяющего все компьютеры сети со специальным соединительным устройством, называемым концентратором. Достоинство данной топологии — высокая надежность работы сети. Обрыв любого проводника отключает только одного абонента.

Для пересылки данных от одного компьютера на другой необходимо знать и указать его адрес. В современных сетях используются три типа адресов: физические, числовые и символьные.

Каждый сетевой адаптер, мост, маршрутизатор и другое сетевое оборудование имеет уникальный цифровой аппаратный адрес, называемый *физическим*, который и используется для адресации в локальной сети. Такой адрес получил название MAC-адреса (Media Access Control — управление доступом к среде). MAC-адрес устройства можно видеть на его тыльной стороне.

Использование *числовых адресов* связано с работой TCP/IP, который является одним из основных протоколов, обеспечивающих доставку информации от источника к адресату. *Символьные адреса* или имена предназначены для запоминания пользователями, поэтому обычно несут смысловую нагрузку. Для локальных сетей символьное имя может иметь краткую форму, например: school627PC002 (школа № 627, персональный компьютер № 002).

В современных сетях для адресации узлов применяют, как правило, одновременно все три схемы. Пользователи адресуются к компьютерам символьными именами, которые автоматически заменяются в передаваемых по сети сообщениях на составные числовые адреса. После доставки сообщения в сеть назначения вместо числового адреса может использоваться аппаратный адрес компьютера или другого устройства в сети.

П-Т Проектное задание

Познакомьтесь с примером (учебным проектом) разработки ЛВС учреждения (http://cisco.novgorod.ru/kurs1.php) и объясните все этапы:

1. Предпроектное обследование, сбор необходимой информации.
2. Выбор (с обоснованием) топологии ЛВС.
3. Выбор (с обоснованием) технологии ЛВС.
4. Составление технического задания на реализацию проекта — техническое задание (пример).
5. Составление схем коммуникаций:

- поэтажные планы с указанием трасс прокладки информационных каналов, точек подключения рабочих мест и размещения коммутационных узлов (пример);
- общая схема кабельной системы с указанием коммутационных узлов, магистральных каналов, вертикальной и горизонтальной кабельной системы (пример1, пример2);
- схема соединений с маркировкой каналов для каждой коммутационной панели (пример).

6. Выбор (с обоснованием) пассивного оборудования кабельной системы.
7. Планирование работ по монтажу и тестированию кабельной системы.
8. Выбор (с обоснованием) активного оборудования ЛВС, мест его размещения — спецификации активного оборудования.
9. Составление таблиц коммутации — таблица коммутации с указанием коммутируемых портов, поддерживаемой технологии для каждого канала передачи данных (пример).
10. Составление общей схемы ЛВС — общая схема ЛВС (рабочая версия).
11. Составление сводного сметного расчета — сводный сметный расчет (пример).

6.3. Интернет-страница и редакторы для ее создания

World Wide Web — глобальная компьютерная сеть — на сегодняшний день содержит миллионы сайтов, на которых размещена всевозможная информация. Для навигации в WWW используют специальные программы — *браузеры*, которые существенно облегчают путешествие по бескрайним просторам сети. Вся информация в браузере отображается в виде web-страниц сайтов. Поддерживая технологию мультимедиа, они объединяют различные виды информации: текст, графику, звук, анимацию и видео. От того, насколько качественно, удобно и красиво сделана та или иная web-страница, во многом зависит ее успех в сети. Пользователю приятно посещать те web-страницы, которые имеют стильное оформление, не отягощены чрезмерно графикой и анимацией, быстро загружаются, обеспечивают интерактивное взаимодействие (эффект диалога) и без помех отображаются в окне web-браузера.

Отсутствие единого стандарта для разработчиков браузеров и web-серверов могло привести к невозможности создать сайт, который одинаково выглядел бы в разных браузерах. Пришлось бы под каждый браузер писать свой сайт. Во избежание этого по инициативе Тима Бернерса-Ли в июле 1994 г. на базе Массачусетского технологического института (Massachusetts Institute of Technlogies — MIT) был создан

World Wide Web Consortium (*W3C*). Основная цель *W3C* — обеспечить как можно большую совместимость ПО web-публикаций. *W3C* не является административным органом, это нечто вроде форума для выработки компромиссных решений в области web-технологий. Консорциум принимает на рассмотрение любые проекты и предложения. Спецификации, разработанные *W3C*, не обязательны для применения, но консорциум ведет работу по их пропаганде. HTML был ратифицирован *W3 Consortium*.

Создать качественную web-страницу непросто, для этого нужны не только навыки дизайнера для красивого и стильного оформления, но и опыт программирования. Сложность и объем программ, требуемых для создания того или иного сайта, зависят от множества факторов, таких как структура, цель, функциональность, обеспечение надежности и система защиты сайта от информационных атак со стороны информации-«прилипалы» (спам, мусор) и вирусов.

Web-технология (технология гипертекста) позволила связать всю совокупность опубликованных в Интернете документов в единую систему — WWW. Гипертекстовый документ содержит *гиперссылки*. Их обычно выделяют цветом и подчеркиванием. Гиперссылки позволяют пользователю немедленно перейти к нужному документу, независимо от места его физического расположения. Это существенно облегчает поиск информации в Интернете.

Использование гипертекстовой разбивки текстового документа в современных информационных системах во многом связано с тем, что гипертекст дает возможность создавать механизм нелинейного просмотра информации. В таких системах данные представляются не в виде непрерывного потока текстовой информации, а набором взаимосвязанных компонентов, переход по которым осуществляется с помощью гиперссылок.

Язык HTML не является собственно языком программирования; это средство описания структуры документа, его стиля и связей его с другими документами. HTML представляет собой упрощенную версию стандартного общего языка разметки — SGML (Standart Generalised Markup Language). Этот язык предназначен для создания других языков разметки, он определяет допустимый набор *тегов* (инструкций языка разметки), их атрибуты и внутреннюю структуру документа. С помощью SGML можно описывать структурированные данные, организовывать информацию, содержащуюся в документах, представлять эту информацию в некотором стандартизованном формате. Но ввиду некоторой своей сложности SGML использовался в основном для описания синтаксиса других языков (наиболее известным из которых является HTML), и немногие приложения работали с SGML-

документами напрямую. Гораздо более простой и удобный, чем SGML, язык HTML позволяет определять оформление элементов документа и имеет некий ограниченный набор тегов, с помощью которых осуществляется процесс разметки. Теги HTML в первую очередь предназначены для управления процессом вывода содержимого документа на экране программы-клиента и определяют способ представления документа, но не его структуру. В качестве элемента гипертекстовой БД, описываемой HTML, используется текстовый файл, который может легко передаваться по сети с использованием протокола HTTP.

Язык разметки документов — это набор специальных инструкций, называемых тегами, которые предназначены для формирования в документах какой-либо структуры и определения отношений между различными элементами этой структуры. Теги языка, или управляющие дескрипторы, в таких документах каким-то образом кодируются, выделяются относительно основного содержимого документа и служат в качестве инструкций для программы, производящей показ содержимого документа на стороне клиента. В самых первых системах для обозначения этих команд использовались символы < и >, внутри которых помещались названия инструкций и их параметры. Сейчас такой способ обозначения тегов является стандартным.

Документ, написанный на языке HTML, представляет собой текст, в который вставлены теги. Они помогают программе просмотра разобраться в том, как должен быть расположен текст на экране, где будут находиться рисунки, хранящиеся в отдельных файлах, и т.д. С их помощью формируются связи с другими интернет-сайтами и web-ресурсами. Создавать такой файл удобнее в специальном редакторе, о котором будет рассказано далее, а просматривать — программой-браузером, например Internet Explorer.

Теги — это последовательности символов, заключенных между знаками < и >. Все, что заключено между тегами <HTML> и </HTML>, является HTML-документом. Любой такой документ состоит из двух частей: первая — *заголовок*, находится между тегами <HEAD> и </HEAD>. Он содержит информацию о документе, которая не выводится на экран. А вот так записывается название странички (сайта): <TITLE>Мой сайт</TITLE>. Вторая часть документа — *тело*, которое выводится на экран программой просмотра; сюда могут входить текст, картинки, видеофрагменты. Эта часть заключается между тегами <BODY> и </BODY>/. Вы уже понимаете, что такое парные теги: первый — открывающий, а второй, с таким же ключевым словом и знаком / — закрывающий. Для создания таких документов можно использовать специальный *web-редактор* (например, FrontPage Express). Однако при этом нужно помнить, что известный текстовый редактор

Word, начиная с Word 97, сам может создавать такие документы, выступая фактически в роли web-редактора.

Итак, создадим HTML-документ с помощью Word:

<HTML>
<HEAD>
<TITLE>МОЙ САЙТ</TITLE>
</HEAD>
<BODY>
ПРИВЕТСТВУЮ ВАС НА МОЕЙ ДОМАШНЕЙ СТРАНИЧКЕ В ИНТЕРНЕТЕ!

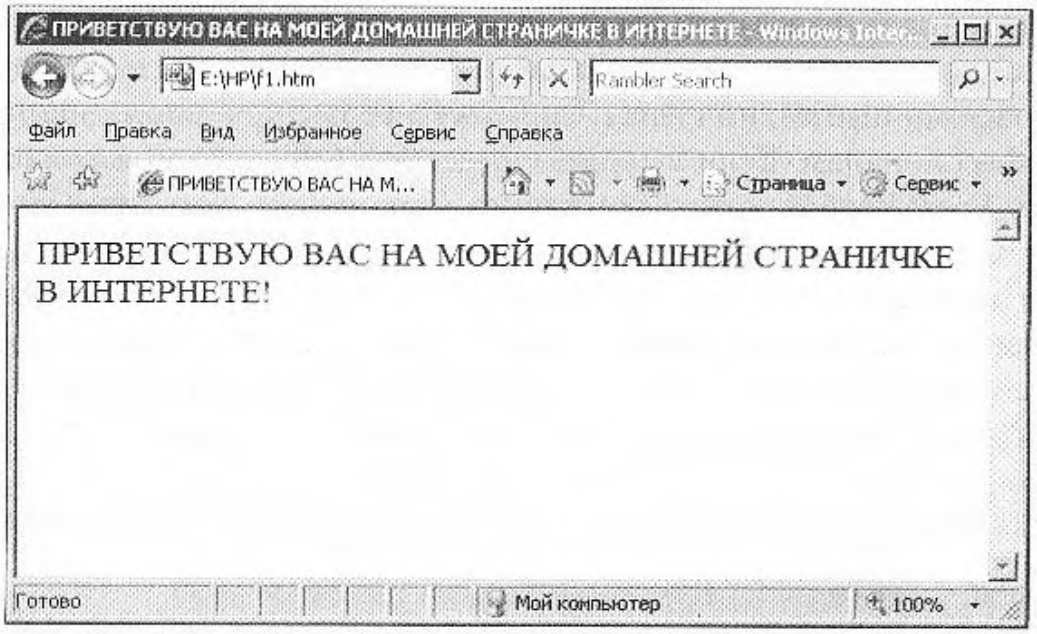


Рис. 6.4. HTML-документ

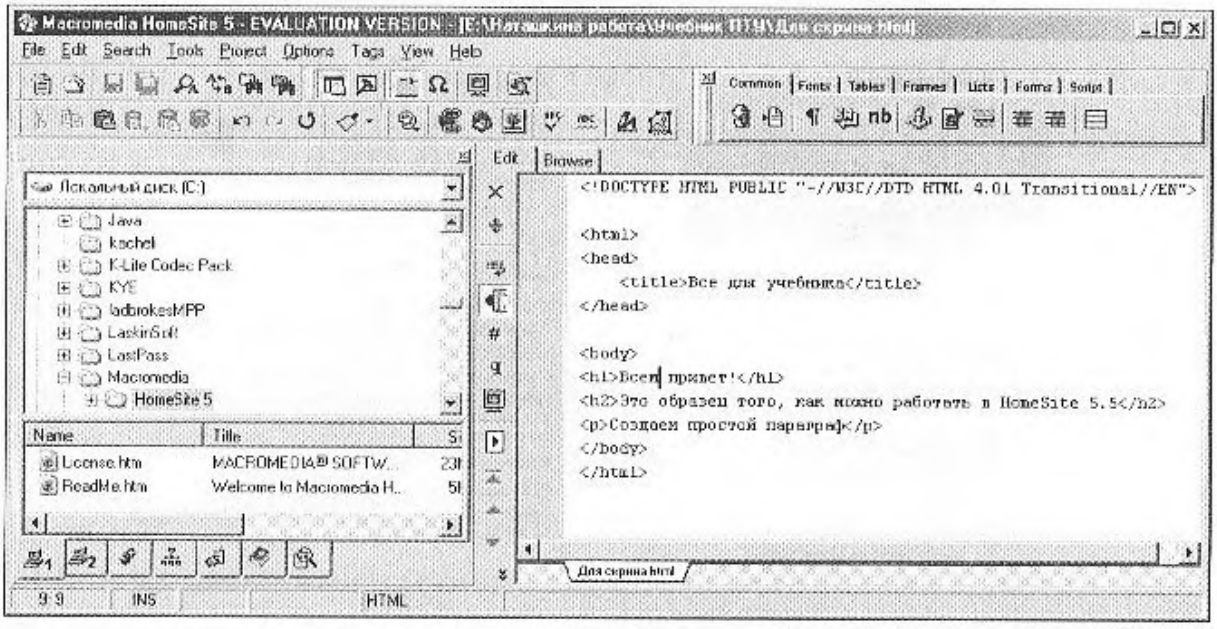


Рис. 6.5. Интерфейс HomeSite

</BODY>
</HTML>

Сохраним этот файл под именем f1.htm, после чего вызовем браузер и выполним **Файл**, **Открыть**. В поле окна увидим надпись, приведенную на рис. 6.4.

Для создания web-страниц вам понадобится текстовый или web-редактор. Надо отметить, что все web-редакторы обязательно содержат и текстовый редактор. web-редакторов сейчас существует большое количество, но самыми мощными признаны три:

1) Macromedia Dreamweaver 3.0;
2) Allaire HomeSite 4.5;
3) Microsoft FrontPage 2000.

Они позволяют создавать целые web-узлы (порталы) и используют все современные технологии. Такие web-редакторы вполне подходят и профессионалам. Если вы пользуетесь MS Office, то можете использовать FrontPage из этого комплекта. Есть также Netscape Composer в составе Netscape Communicator. Можно также воспользоваться демонстрационной версией web-редактора Macromedia HomeSite 5.5, получив ее на официальном сайте компании *Adobe*: www.adobe.com. С помощью браузера следует зайти на указанный сайт в раздел Downloads/Trial Downloads и выбрать Macromedia, HomeSite 5.5 (рис. 6.5).

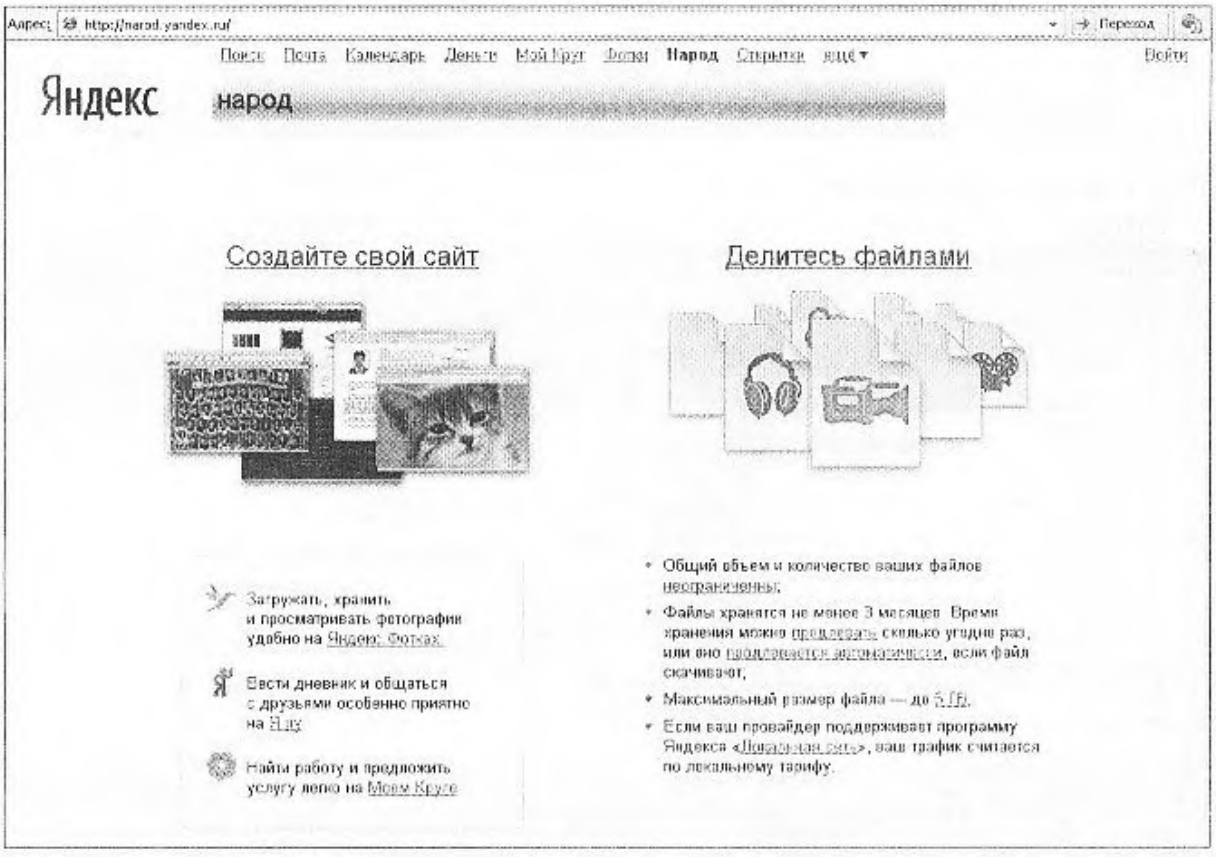


Рис. 6.6. Конструктор сайтов

Далее необходимо зарегистрироваться на странице загрузки, чтобы получить возможность загрузить демоверсию программы для ее свободного использования в течение 30 дней.

Можно воспользоваться русской версией FrontPage Express 2.0 из локализованного Internet Explorer 5.0 — это сокращенный вариант FrontPage 98, но очень удобен для начала. На русском языке также известен web-редактор Visual Workshop (http://www.snkey.net/download/products/index.html).

Современные сервисы в Интернете предлагают готовые *конструкторы сайтов*. Рассмотрим их на примере конструктора сайтов «Мой сайт» (www.narod.yandex.ru). С его помощью можно получить место под свой сайт (домашнюю страницу) и за 5 мин создать свой web-сайт с картинками, фотографиями и пр.

Процесс конструирования сайта включает в себя три этапа:

1) вход в конструктор сайта (рис. 6.6);
2) регистрация. Назначение имени сайта (рис. 6.7—6.9);
3) конструирование сайта с использованием заготовок (рис. 6.10, 6.11).

Для создания рукотворного сайта, единственного в своем роде, требуются иные web-инструменты.

Помимо HTML и конструкторов сайтов можно пользоваться и другими средствами программирования web-страниц. Одним из средств программирования сайтов является CGI (Common Gateway

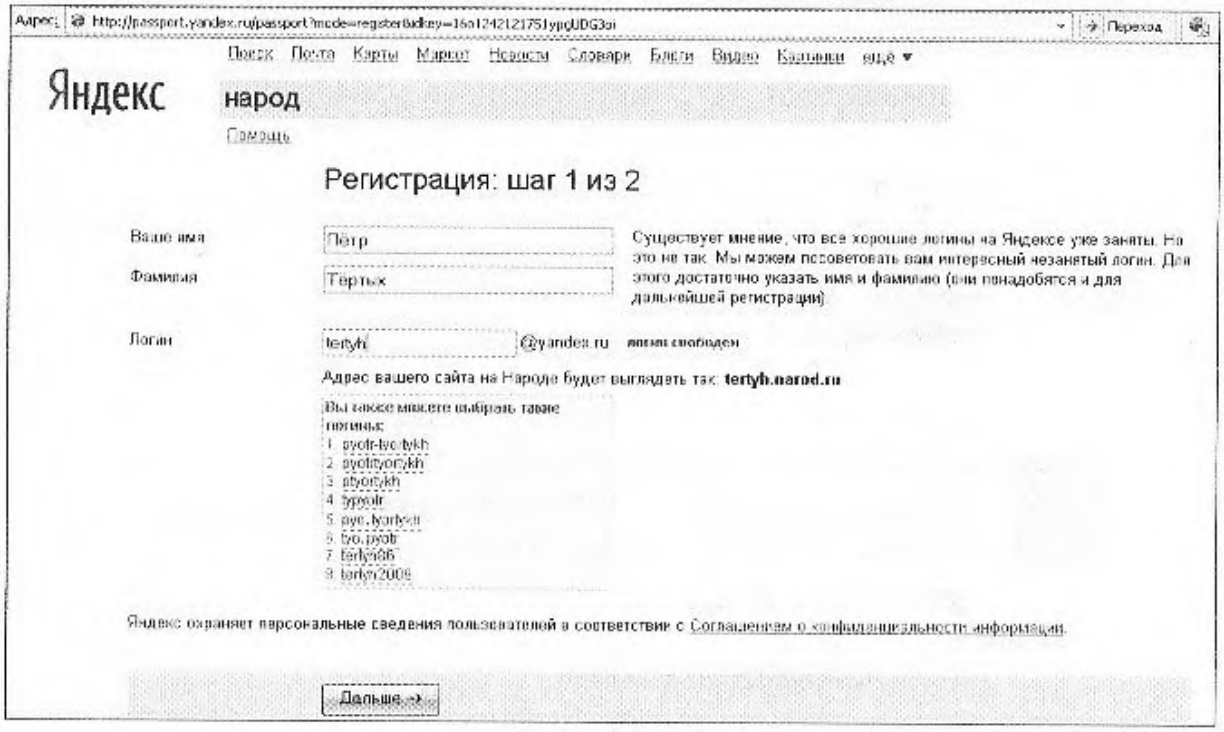


Рис. 6.7. Регистрация на сайте: шаг 1

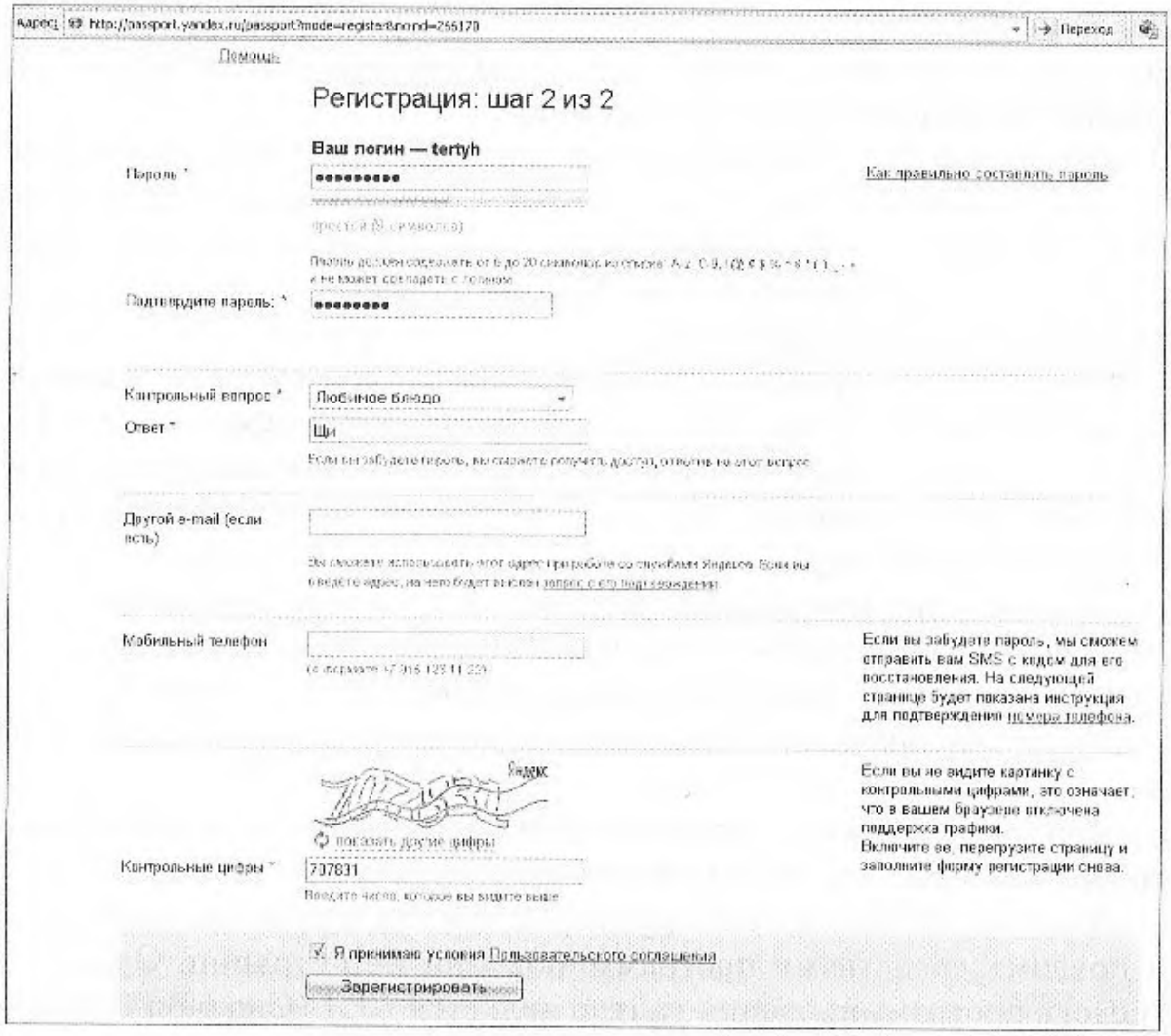


Рис. 6.8. Регистрация на сайте: шаг 2



Рис. 6.9. Успешное завершение регистрации

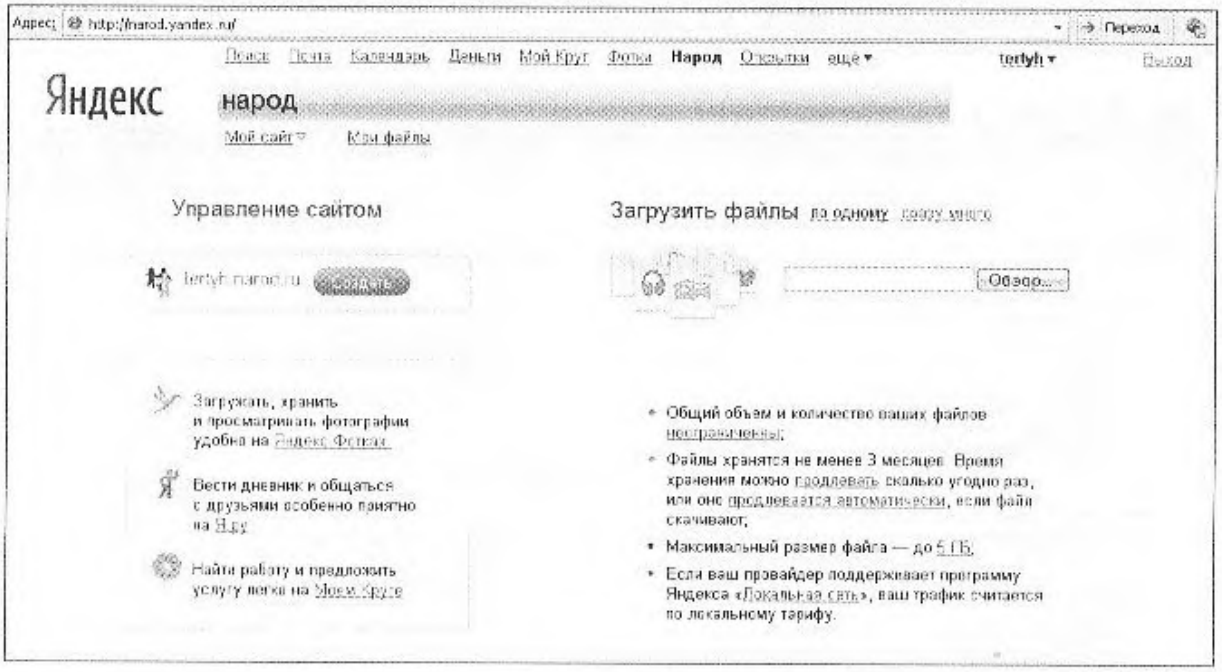


Рис. 6.10. Страница управления сайтом

Interface — общий интерфейс шлюзов). Принцип CGI прост и естественен: программа на стороне сервера принимает некие данные, обрабатывает их и выдает ответ в виде web-страницы. Это очень частое явление на сегодняшний день, и порой даже не подозреваешь, что имеешь дело не с обычной страницей, а с результатом работы CGI-скрипта (кода). Область использования CGI широка, а цели разнообразны. Язык программирования, на котором пишется скрипт, должен быть гибким и в то же время мощным, так как предстоит не только постоянная работа с файлами, но и с БД, ОС и web-сервером. Самыми подходящими языками для CGI-программирования являются Си (C++) и Perl. Последний наиболее часто используется для этих целей из-за своей простоты, надежности и широты возможностей.

Язык Си по праву считается универсальным и нашел свое место в среде CGI. Его применение естественно в родной стихии — Unix, на основе которой в большинстве случаев и строится web-узел.

Для CGI-программирования используется также *язык C++*, который подразумевает создание и уничтожение неких объектов, реализуемых через классы, — по сути пользовательских типов данных. Естественно создать, например, класс Guestbook или DataBaseEntry, который содержал бы необходимые переменные и функции. Стоит заметить, что почти все новые языки программирования (Java, JavaScript, Perl) содержат конструкции, схожие с конструкциями Си или же являющиеся их точными копиями. Поэтому рекомендуется сначала изучить именно этот фундаментальный язык.

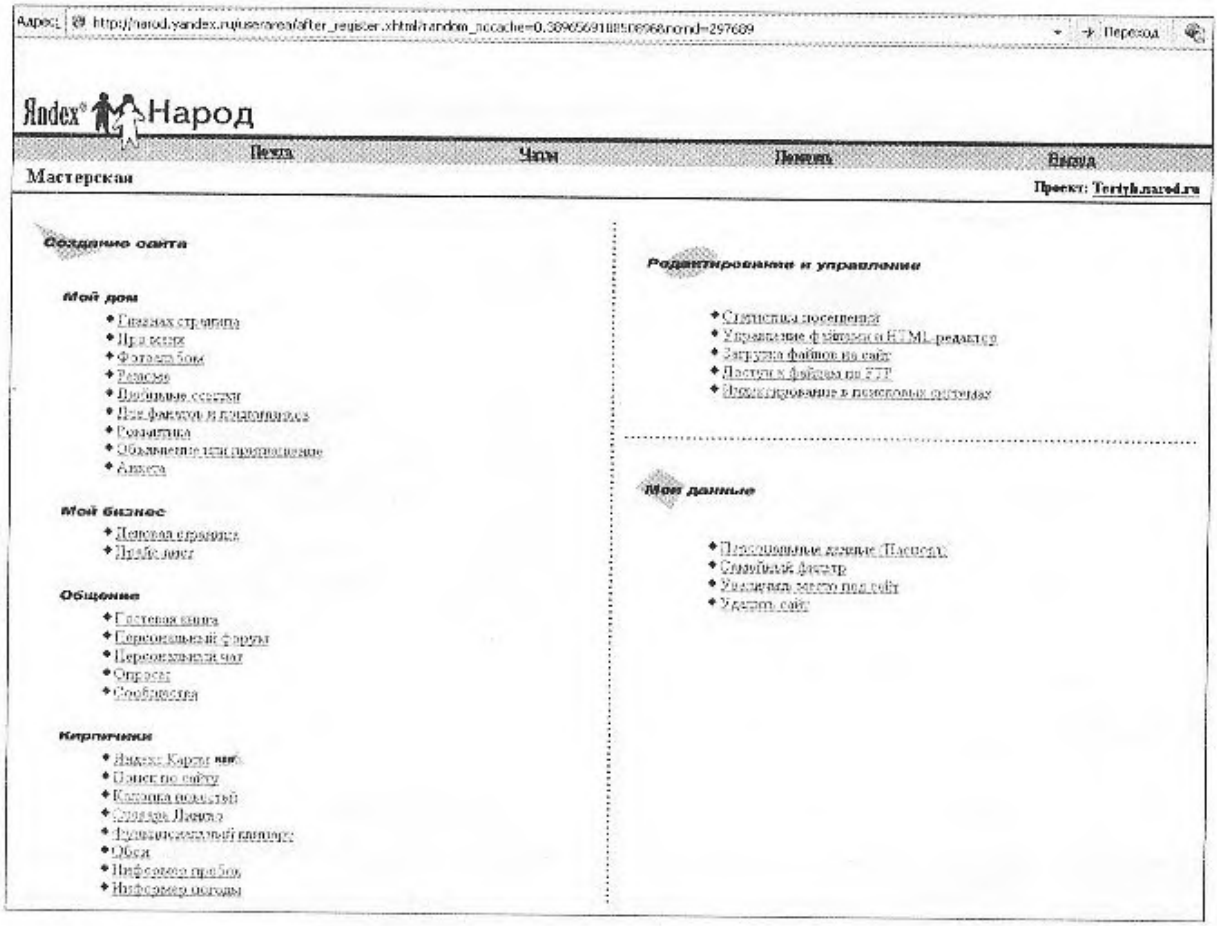


Рис. 6.11. Окно «Мастерская»

Язык Perl уникален. Во-первых, в нем отсутствуют типы данных, что благоприятно воздействует на написание CGI-скриптов. Во-вторых, очень много конструкций взято из Си. Надо отметить, что в Perl часто используются регулярные выражения, похожие на те, что применяются в сценариях Unix Shell. Имеется удобный цикл foreach и красивые, по сути, модификаторы выражений. Perl располагает стандартной библиотекой, в которой есть функции поиска, работы с web-сервером, графикой и т. п. Из этого можно сделать вывод, что для программиста, знакомого с Unix, не составит труда изучить Perl. Для написания такой программы достаточно уметь работать с файлами из Perl и, немного, с переменными окружения web-сервера.

Вопросы и задания

1. Что такое web-технология?
2. Сравните браузеры для разных ОС, используя цветную вклейку, рис. 7, 8..
3. Что понимают под web-страницей?
4. Что представляет собой язык гипертекстовой разметки HTML?
5. Создайте web-страницу в редакторе Word.

6.4. Личные сетевые сервисы в Интернете

Личные сетевые сервисы принадлежат конкретному пользователю, защищены логином и паролем, доступ к ним осуществляется только персонально владельцем сервиса. Примером такого сервиса может служить личный ящик электронной почты, личный сайт или раздел (блог), личный телефонный номер и пароль для его использования, личная страница на сайте дистанционного обучения или дистанционного тестирования. Все личные сервисы в Интернете требуют *персонализации*, т. е. регистрации со стороны пользователя, по итогам чего пользователь получает личный логин и пароль и пр.

Личная информационная деятельность в Интернете имеет в основном поисковый и коммуникативный характер. Рассмотрим коммуникативный вид сетевой деятельности на примере электронной почты, а поисковый — на примере работы пользователя с поисковыми серверами.

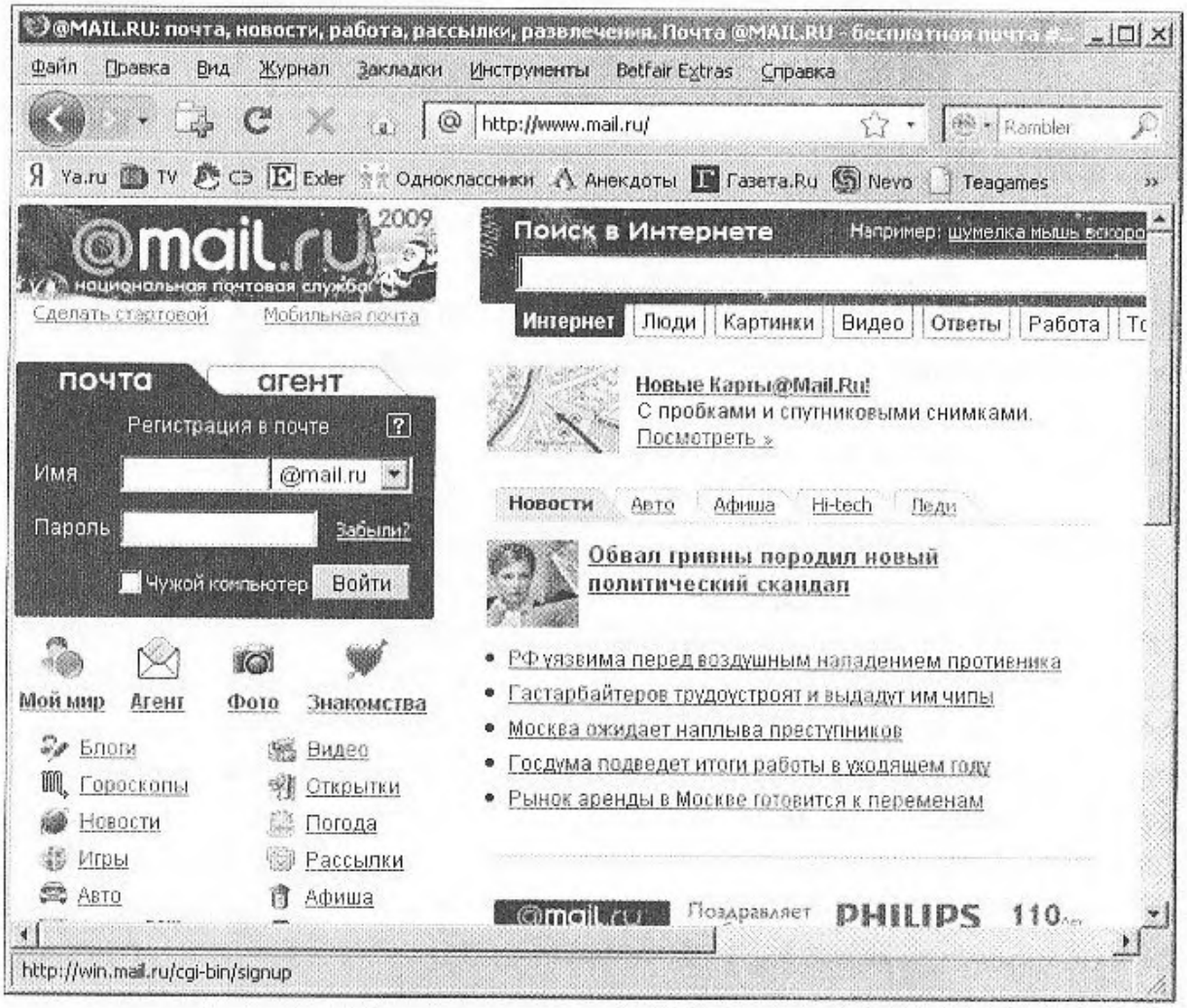


Рис. 6.12. Сайт электронной почты на портале www.mail.ru

Для того чтобы с кем то переписываться по электронной почте, нужно иметь свой почтовый ящик. Его можно завести на любом поисковом или почтовом сервере. Теперь на многих серверах общего назначения имеется почта.

Выберем наиболее известный почтовый сервер www.mail.ru и войдем на него (рис. 6.12).

Сначала нужно зарегистрироваться, для чего вызвать услугу **Регистрация в почте**. В появившемся окне необходимо заполнить все поля, отмеченные «*», иначе регистрация не состоится (рис. 6.13). Главные поля — это *логин* (т.е. ваше уникальное имя) и *пароль*, без которого никто не сможет войти в ваш почтовый ящик.

Мы выбрали логин marsolov67 и пароль, а также ввели другие данные, многие из которых не являются обязательными. Система сообщит нам, что мы зарегистрированы (рис. 6.14).

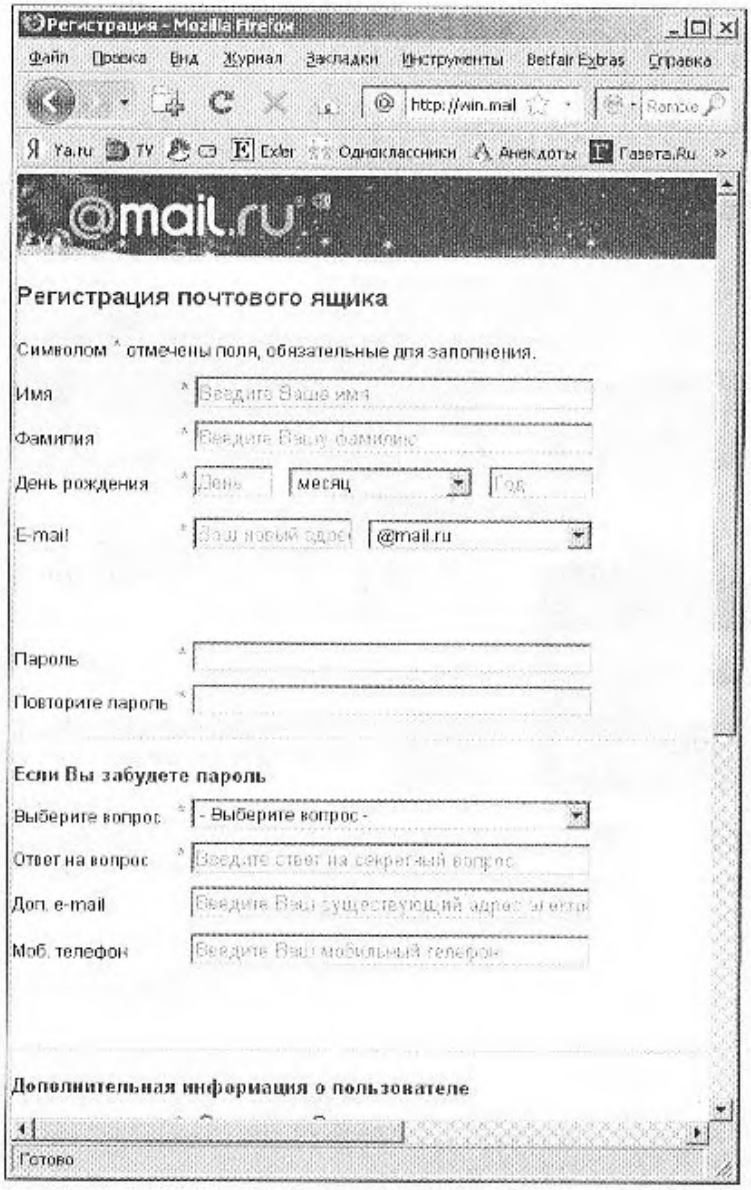


Рис. 6.13. Регистрация почтового ящика

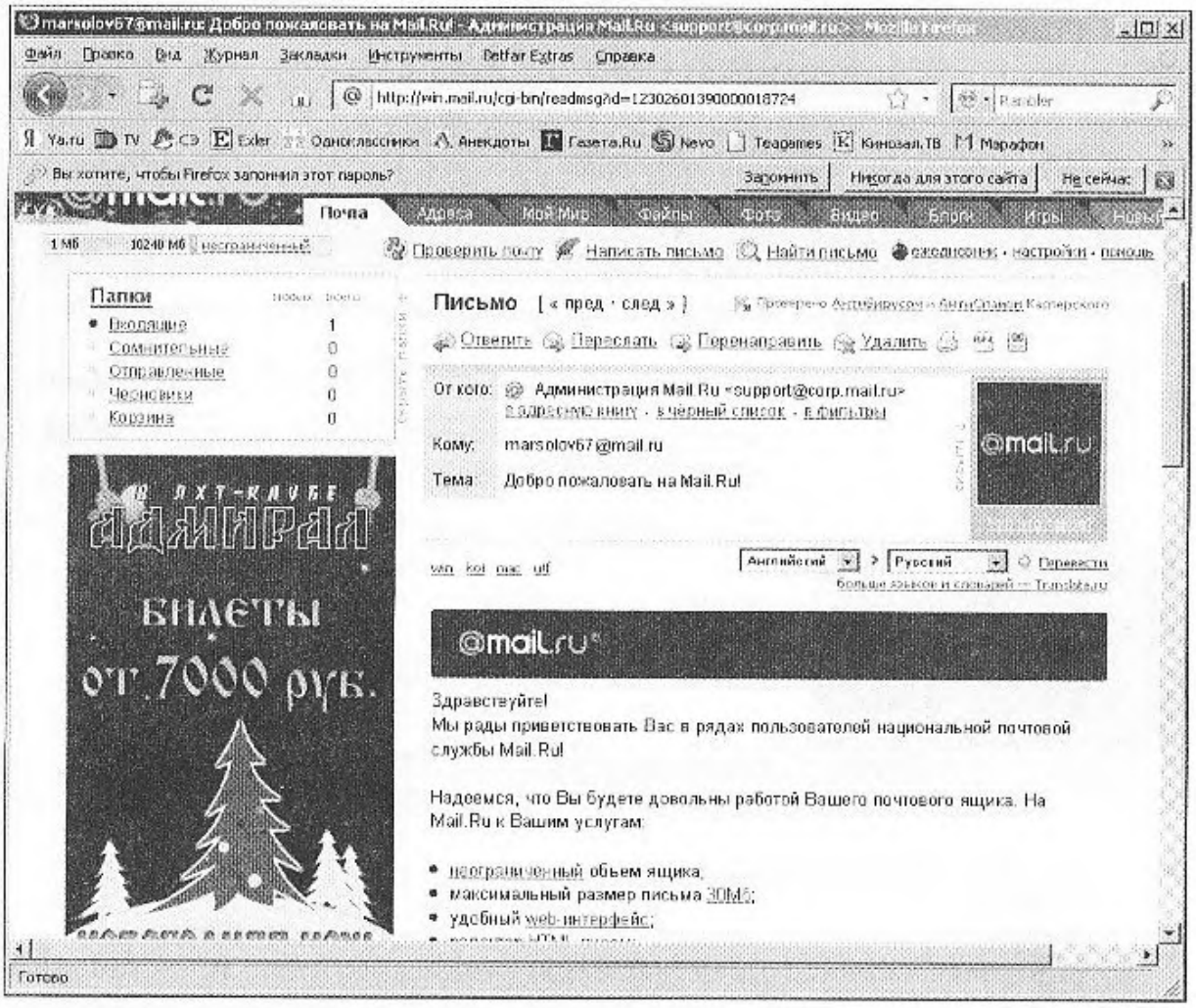


Рис. 6.14. Просмотр письма в почтовом ящике

Теперь наш почтовый адрес marsolov67@mail.ru. Мы уже можем участвовать в переписке. Например, у нас есть знакомый, и мы знаем его адрес: lesov@mn.ru. Набираем в строке адреса нашего браузера www.mail.ru и в соответствующие окна вводим свой логин и пароль.

Выбираем **Написать письмо** (рис. 6.15). Обязательно нужно заполнить поле «кому», т. е. указать адрес вашего друга (в данном случае lesov@mn.ru), поля «копия» и «скрытая» необязательны. Особо следует отметить общепринятые нормы заполнения поля «тема». Лучше указывать кратко основную идею сообщения, это поможет и вам, и адресату более быстро ориентироваться в почтовых сообщениях. К письму можно прикрепить файлы объемом до 15 Мбайт (тексты, фото, музыку и т. д.), для чего имеются окна установки файла в нижней части окна оформления письма. Написав письмо, вызовем услугу **Отправить** (рис. 6.16). Система обязательно сообщит об успешности отправки или ошибках. При желании можно сохранять копии отправленных писем.

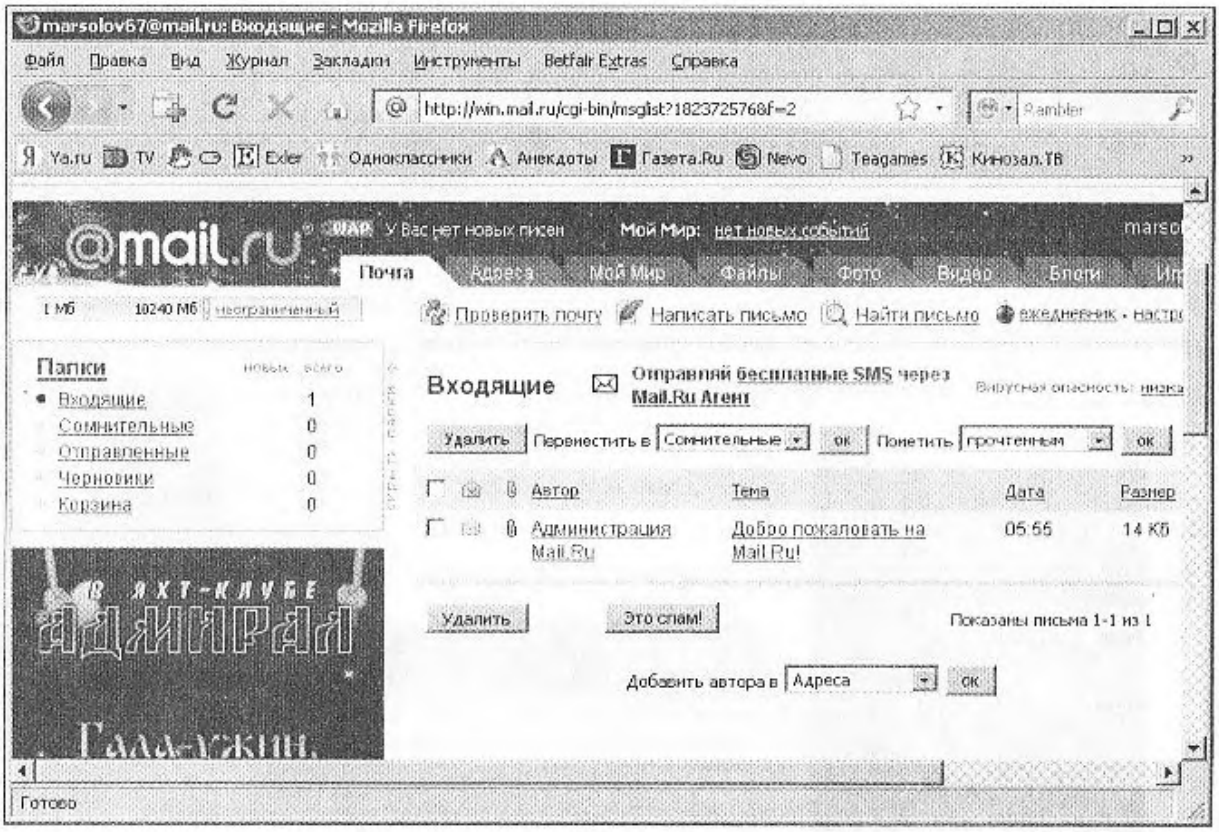


Рис. 6.15. Просмотр почтового ящика

В почтовом сервисе предлагаются папки сортировки сообщений: «Входящие, «Сомнительные», «Отправленные», «Черновики», «Корзина» (рис. 6.17). В этих папках можно хранить письма, а по желанию — очищать их. Необходимо отметить, что непрочитанные письма отмечаются более темным цветом.

Используя электронную почту, часто невозможно передать файл большого размера. Как правило, современные почтовые серверы накладывают ограничение на размер пересылаемого файла — 5 или 10 Мбайт. Для обмена большими файлами многими провайдерами социальных сервисов в Интернете предоставляется услуга *файлообмена*. В качестве примера можно воспользоваться службой обмена файлами портала Mail.ru — http://files.mail.ru/ (рис. 6.18).

Каждый пользователь Интернета использует **поисковые сервисы**. Для выхода в Интернет потребуется специальная программа-браузер. Самыми популярными на сегодняшний день программами-браузерами являются Internet Explorer и Netscape Navigator. Так как Internet Explorer по умолчанию входит в ОС Windows, этот браузер более распространен, и работать будем именно с ним. Вызвать его можно из меню **Пуск**, **Программы**, **Internet Explorer** (рис. 6.19).

Нажав на стрелку открытия списка адресов, вы увидите адреса сайтов, по которым уже осуществлялся вход с данного компьютера.

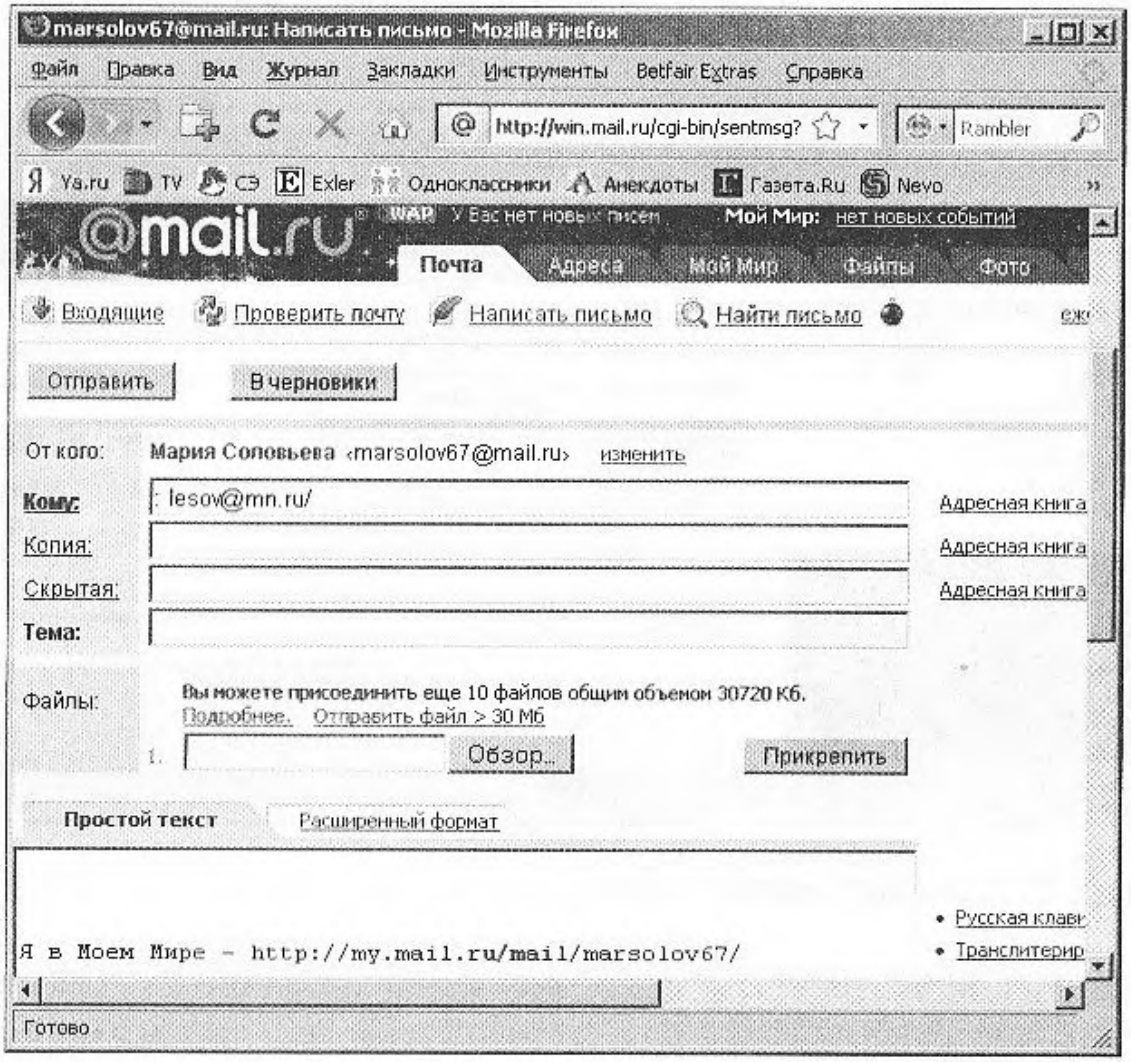


Рис. 6.16. Создание письма

Если вам нужен один из них, достаточно щелкнуть по нему мышкой. Если же вам необходим новый адрес, то следует набрать его в строке адреса и нажать **ENTER**. Например, для входа на поисковый сервер Yandex требуется набрать адрес www.yandex.ru (этот адрес называют еще URL), в результате вы попадете на главную страницу искомого сайта (рис. 6.20).

Развевающийся флаг в правом верхнем углу экрана говорит о том, что работа с Интернетом происходит и сетевое соединение с Интернетом компьютера не прервано. В верхней части панели браузера находится меню. Кнопки со стрелками вправо и влево означают перемещения вперед и назад соответственно, **Обновить** — перезагрузку данной страницы; **Домой** — возврат на стартовую или домашнюю страницу; **Поиск** — поиск в Интернете нужной информации; **Избранное** — вызов наиболее понравившихся страниц, адреса которых вы запомнили; **Печать** — вывод на принтер понравившихся материалов, **Размер** — выбор размеров шрифта, **Кодировка** — выбор вида кодировки. Среди поисковых серверов популяр-

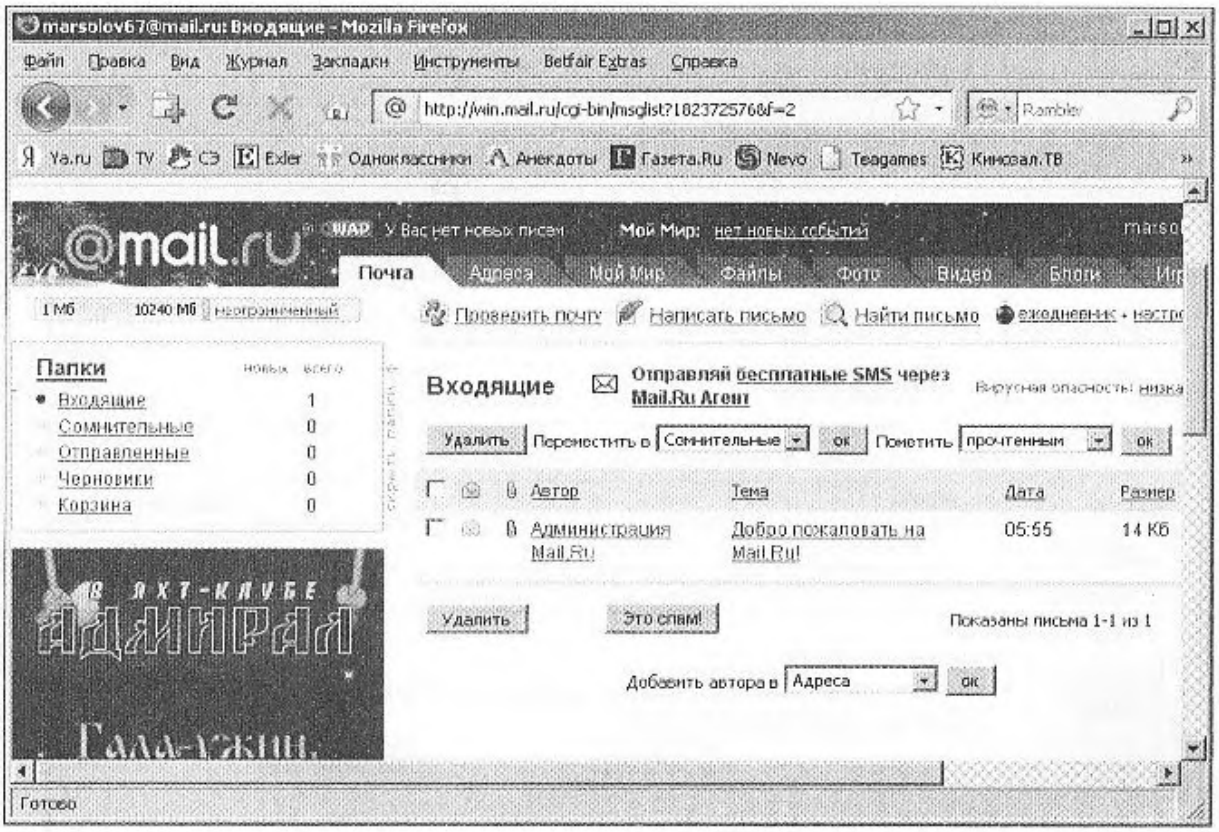


Рис. 6.17. Структура почтового ящика

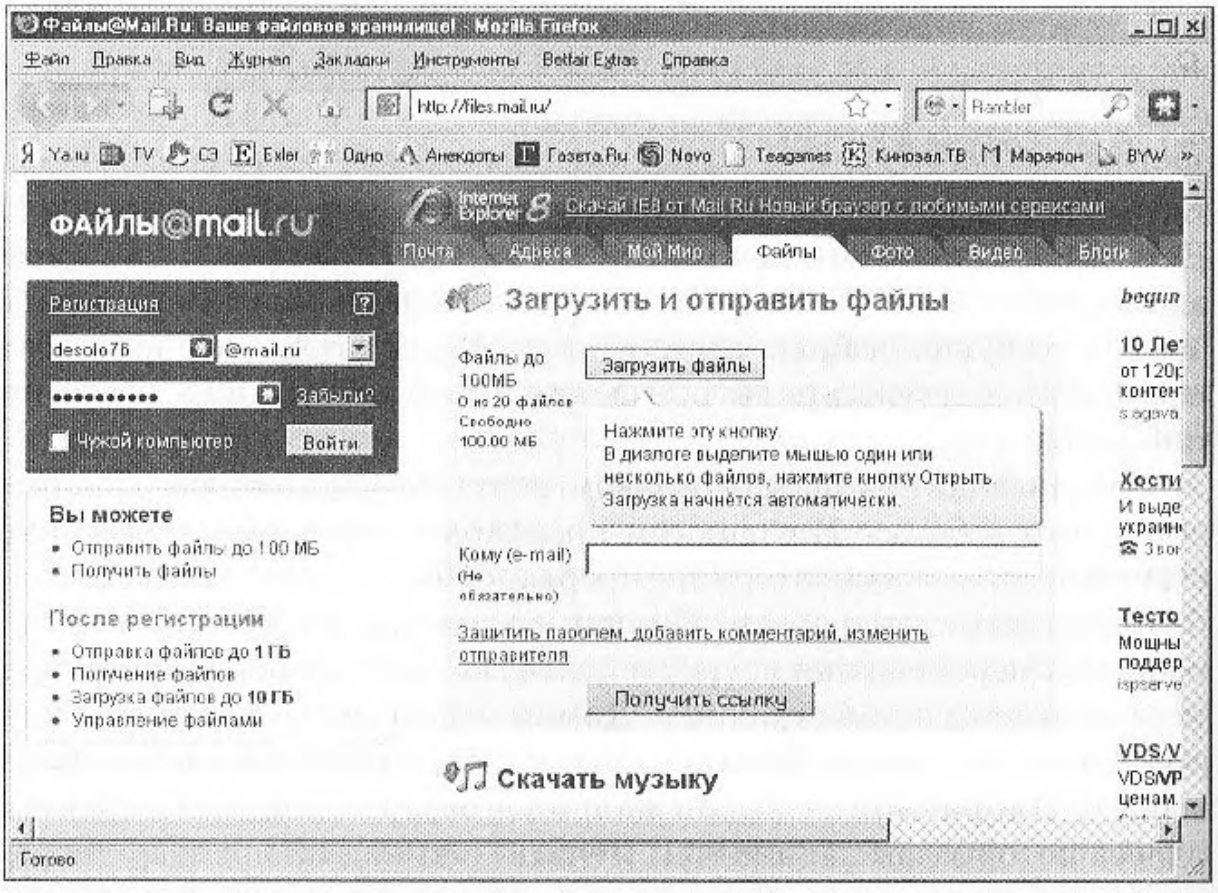


Рис. 6.18. Обмен файлами на портале www.mail.ru

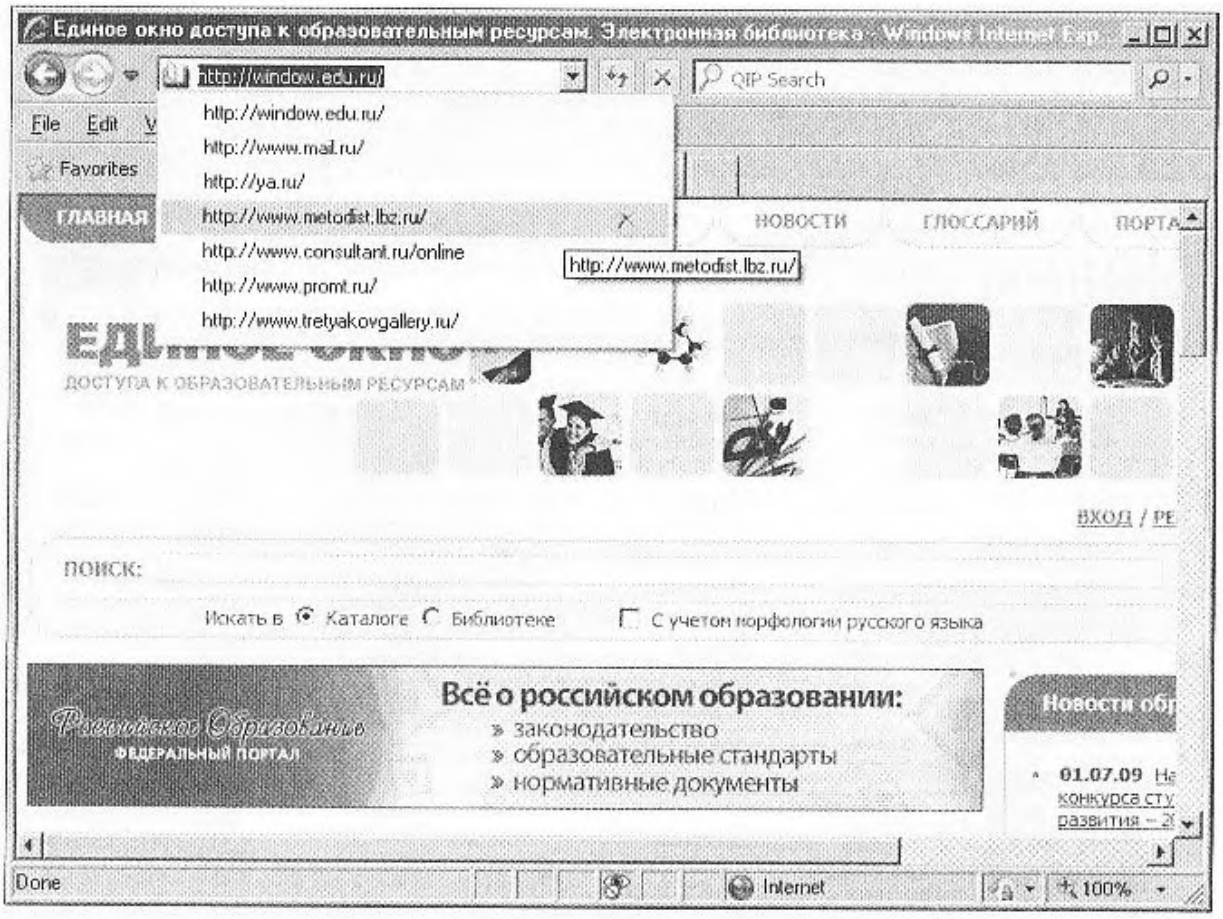


Рис. 6.19. Окно Internet Explorer

ны www.google.ru, http://list.mail.ru/, www.yahoo.com, www.aport.ru и др. Современные поисковые сервисы включают в свою сферу новости, репортажи, телепередачи, интернет-СМИ. Примером такого расширенного поисково-информационного сервиса является www.km.ru. Информацию о последних новостях можно получить на сайтах www.utro.ru, www.news.ru, www.gaseta.ru, информацию о новостях бизнеса — на www.rbc.ru, о погоде на www.pogoda.ru. Не только новости, но и видеорепортажи доступны для пользователя Интернета (рис. 6.21).

Очень интересная возможность — использование *web-камер*, которые с определенной периодичностью, измеряемой иногда секундами, передают изображение по линиям Интернета.

Можно посмотреть на Москву, вид которой получен от web-камеры, установленной на 13 этаже главного здания МГУ в реальном масштабе времени (http://webcam.mnc.ru/). Такие камеры установлены по всему миру и позволяют, не выходя из дома, посмотреть интернет-репортаж о Ниагарском водопаде, шторме у побережья Австралии, городской жизни Парижа и многое другое (рис. 6.22).

Используя Интернет, можно высказать свое мнение на разных форумах, поучаствовать в электронных конференциях, пообщаться

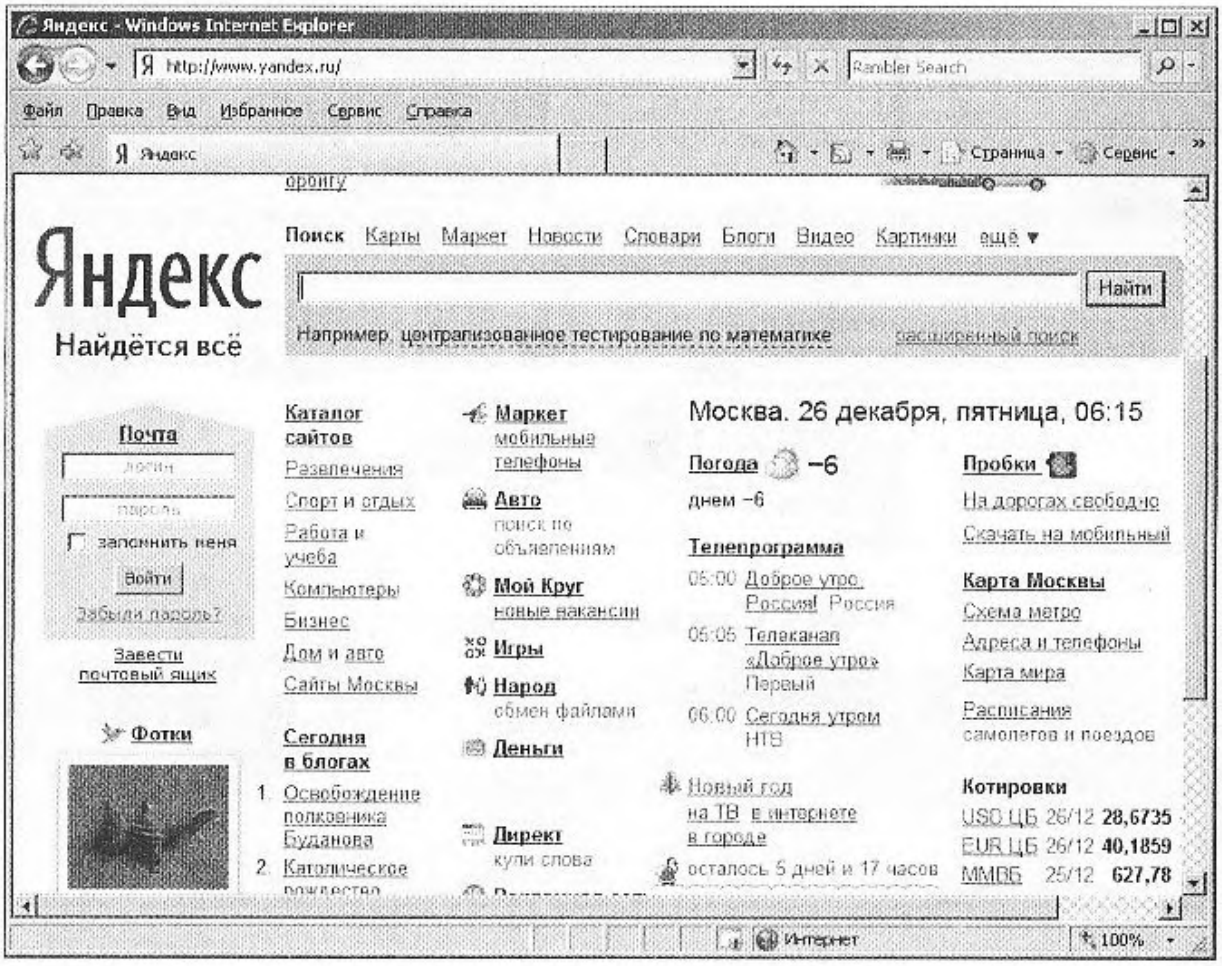


Рис. 6.20. Вход на поисковый сервер Yandex

с друзьями в чатах, сделать покупки, написать письма или послать виртуальные открытки друзьям, послушать хорошую музыку, совершить путешествие по городам и странам и т. д.

С помощью любого поискового сервера Интернета (Yandex, Rambler, Google и т. п.) можно размещать свои *личные объявления*. Для этого надо задать поисковый запрос из ключевых слов, например: «Электронная доска объявлений». Нажав на одну из появившихся ссылок, в частности «ЛесРук» (www.lesruk.ru), вы увидите картину, приведенную на рис. 6.23.

Все объявления сгруппированы по различным категориям. Если вы хотите сами оставить на доске какую-либо информацию, после регистрации следует нажать на строку **Добавить объявление**. На экране появится страница, приведенная на рис. 6.24.

Затем необходимо выбрать нужный раздел, написать в появившемся окне текст сообщения и нажать кнопку **Добавить**. Через несколько секунд ваше объявление будет размещено на электронной доске и станет доступным огромному числу пользователей Всемирной паутины.



Рис. 6.21. Информационный сервис www.vesti.ru

Одна из форм личного сервиса — регистрация и предоставление тезисов для *конференции*, в том числе видео. Существуют конференции самой разнообразной тематики: общественные, технические, медицинские, образовательные, научные и т.д. Примером является известная конференция «Информационные технологии в образовании» (http://ito.su), приведенная на рис. 6.25.

Популярным личным сетевым сервисом стали личные сайты — *блоги*. Можно создать свой блог, например, зарегистрировавшись на сайте www.pedsovet.org (рис. 6.26).

Важнейшее место в личных сетевых сервисах занимает *дистанционное обучение*. Оно позволяет выбрать свою траекторию обучения, а среда сформирует для вас журнал успеваемости. Зарегистрировавшись на сайте дистанционного обучения свободного доступа «Сетевой практикум по информатике» (http://webpractice.cm.ru), можно изучать те разделы, которые наиболее интересны лично вам (рис. 6.27).

Интернет дает возможность удаленно оценить знания, проанализировать личностные качества, профориентационные особенности с помощью сайтов, предоставляющих услуги *онлайн-тестирования*. В ка-

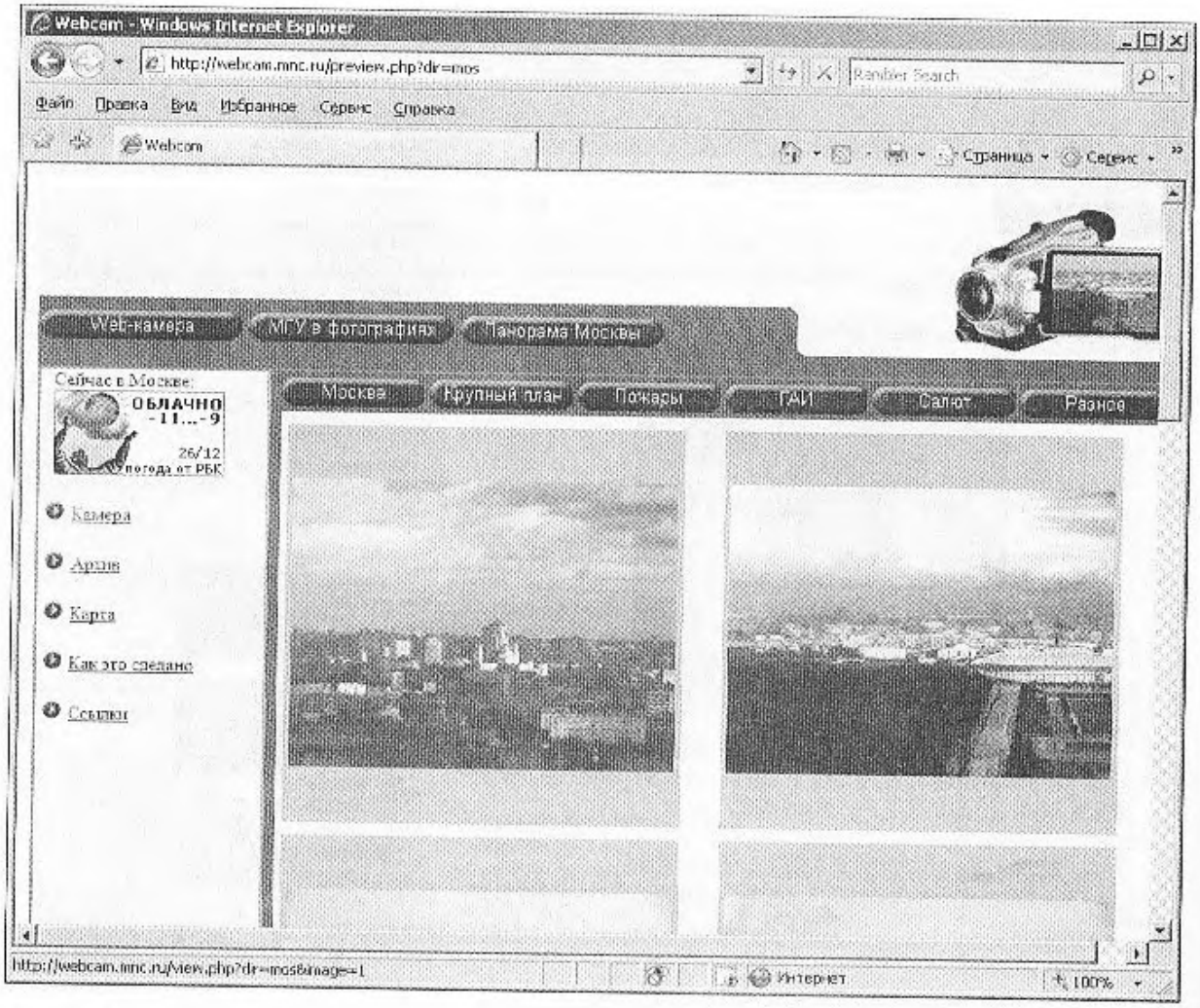


Рис. 6.22. Сайт WebCam

честве примера можно попробовать пройти тесты ЕГЭ прошлых лет на сайте http://abiturcenter.ru/testi/index_n.php (рис. 6.28).

Вопросы и задания

1. Какие сетевые сервисы дистанционного обучения вы используете?
2. Используя свой почтовый ящик и услугу файлообмена, передайте по нему подборку фотографий.
3. Создайте свой блог.
4. Найдите новости о последних достижениях в области информационных технологий.
5. Пройдите онлайн-тестирование.

6.5. Коллективные сетевые сервисы в Интернете

В структуре WWW-сервиса для широкого круга пользователей реализованы разнообразные сетевые услуги коллективного использо-

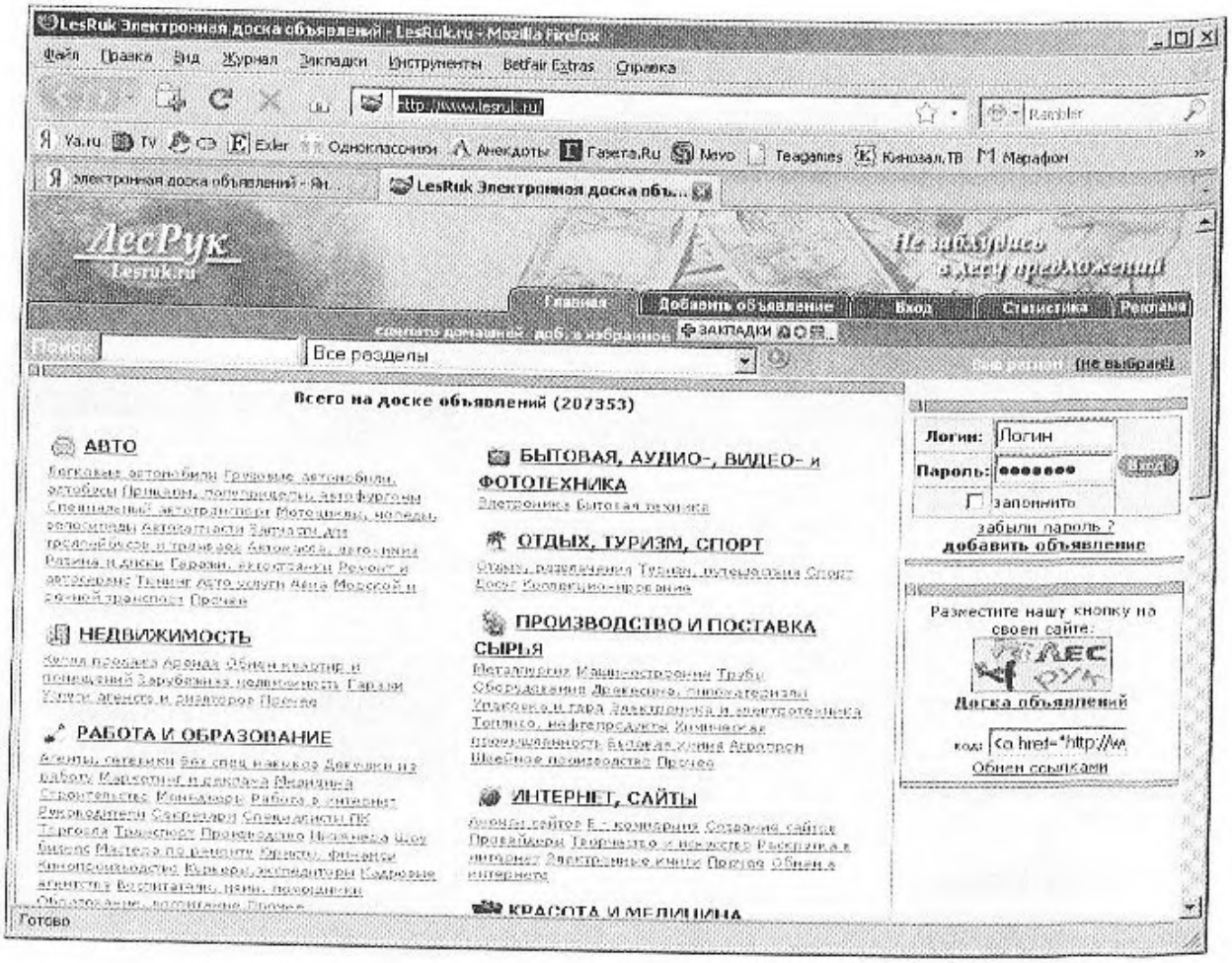


Рис. 6.23. Доска объявлений «ЛесРук»

вания. В таких сервисах могут одновременно принимать участие несколько пользователей.

Современные средства удаленного взаимодействия — *телекоммуникации* — позволяют очень эффективно обмениваться различной информацией. В настоящее время уже стало нормой проводить через Интернет такие средства удаленного взаимодействия, как онлайн-опросы, а с помощью видеокамер — телеконференции.

Рассмотрим организацию *коллективного обсуждения* в сети в режиме реального времени на примере портала Mail.ru (http://www.chat.ru).

Для того чтобы общаться в **чате** (от англ. *chat* — разговор), нужно зарегистрироваться на сайте, предоставляющем данную услугу, потом набрать свои сообщения и отправить их в режиме постоянного подключения к Интернету. На экране постоянно будут идти обновления — это приходят ваши сообщения и сообщения других посетителей чата. На них можно отвечать, игнорировать их, ставить разные смайлики, с особо понравившимся пользователем можно уйти в приват-чат и говорить наедине. Пример чата представлен на странице (http://chat..chat..ru/) (рис. 6.29).

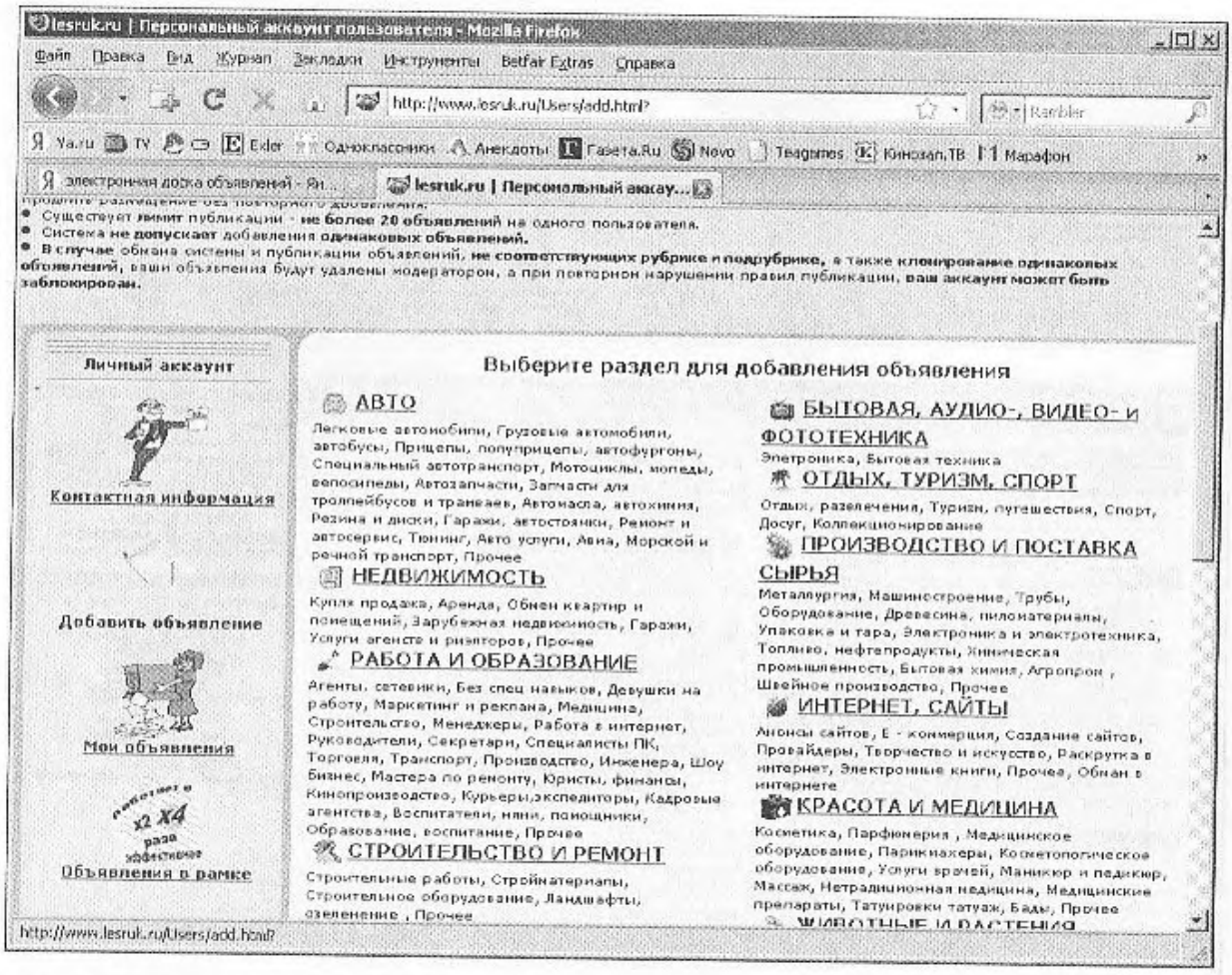


Рис. 6.24. Добавление объявления

Служба мгновенных сообщений (Instant Messaging Service — IMS) — класс программ, предназначенных для обмена сообщениями через Интернет в реальном времени. Передаваться могут текстовые сообщения, звуковые сигналы, картинки, видео. С помощью службы мгновенных сообщений SMS передается на номер телефона вашего абонента, причем совершенно бесплатно. Они могут также применяться для организации телеконференций. Для этого вида коммуникации необходима клиентская программа, так называемый *мессенджер* (от англ. *messenger* — курьер).

Широкому кругу пользователей известны некоторые популярные сети обмена сообщениями, такие как ICQ, MSN, Yahoo!. Каждая из этих сетей разработана отдельной группой разработчиков, имеет отдельный сервер, отличается своими правилами и особенностями. Между различными сетями обычно нет никакой взаимосвязи. Таким образом, пользователь сети ICQ не может связаться с пользователем сети MSN. Однако ничто не мешает быть одновременно пользователем нескольких сетей.

Портал Mail.ru имеет свою службу мгновенных сообщений — Mail.ru Агент (рис. 6.30), причем пользователю предоставляется возможность

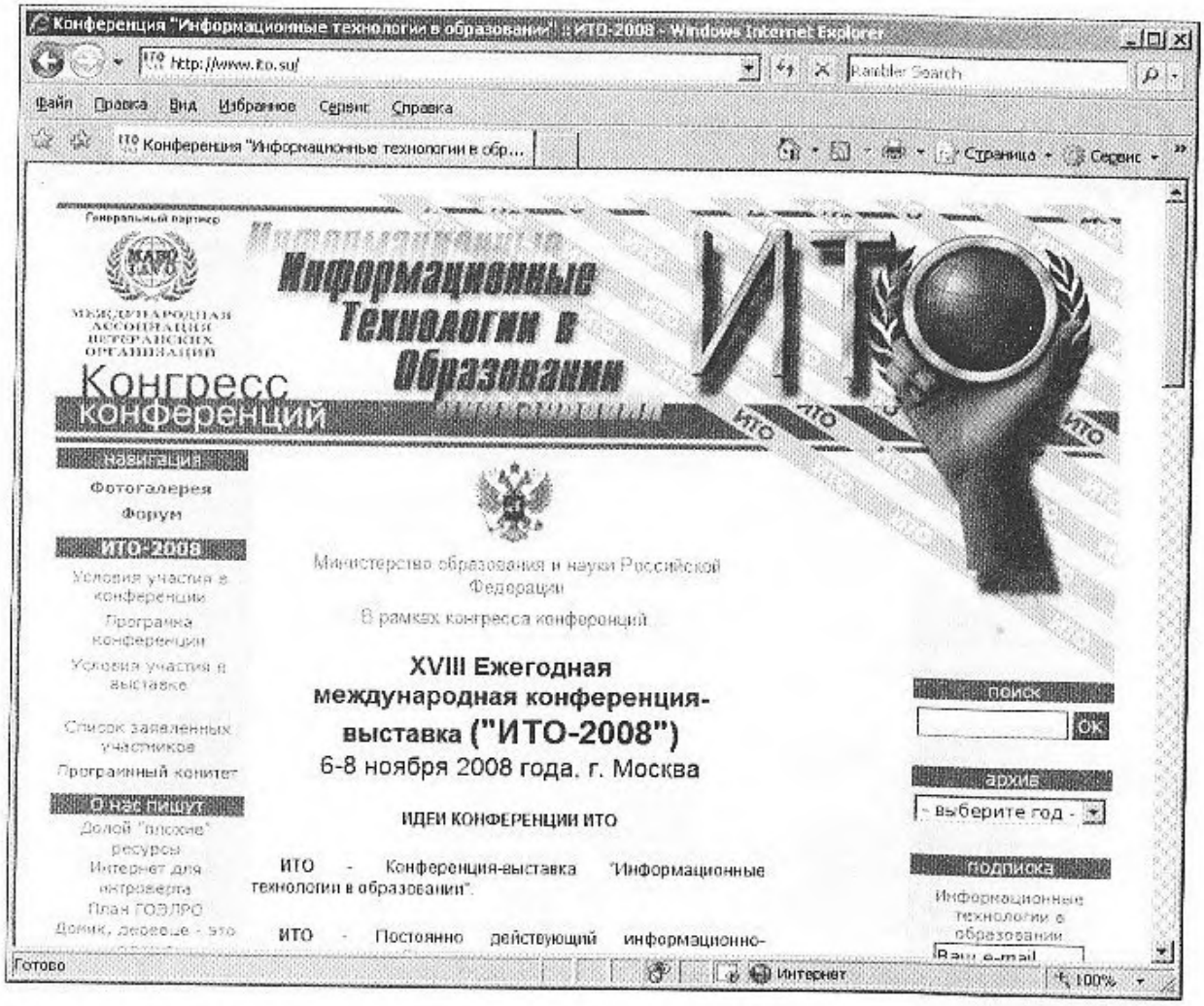


Рис. 6.25. Сайт конференции «Информационные технологии в образовании»

установить клиентскую программу как на компьютере, так и на своем мобильном телефоне или коммуникаторе. Mail.ru Агент (http://agent.mail.ru/ru/) совместим с наиболее популярной сетью ICQ (http://www.icq.com/) (рис. 6.31).

Мобильный Mail.ru Агент кроме всего прочего предоставляет пользователю возможность работать со своей электронной почтой с помощью мобильного телефона.

Форум — это коллективная форма общения на сайте. Каждое сообщение на форуме имеет своего автора, тему и собственное содержание. Как правило, форумы подразделяются на темы, и каждый ваш ответ будет частью общего открытого обсуждения определенной темы. Круг тем форумов ничем не ограничен. Еще одна особенность форума состоит в том, что не обязательно отвечать в режиме реального времени. Темы и ответы хранятся в течение неограниченного периода. Есть форумы, существующие по многу лет, и несколько лет может идти обсуждение одной темы. В качестве примера приведем форум «Информационные технологии в образовании» на сайте http://pedsovet.org/forum/.



Рис. 6.26. Создание блога на сайте www.pedsovet.org

Взаимодействие людей в Интернете помимо общения по интересам имеет и деловой характер, связанный с работой или ее поиском, т.е. это **взаимодействие с потенциальным работодателем**.

В условиях современного рынка труда *резюме*, т.е. сведения о соискателе работы, имеют очень большое значение, поскольку первое впечатление о соискателе у работодателя складывается как раз на основе информации, содержащейся в резюме. Конечно, можно написать резюме в произвольной форме и послать либо электронной, либо обычной почтой. Но лучше все-таки следовать определенному шаблону. Если вы ищете работу с помощью Интернета, то на специализированных сайтах, например www.job.ru, www.superjob.ru, www.rabota.ru, форма резюме вам предлагается, остается только заполнить поля. В качестве примера обратимся к сайту www.job.ru (рис. 6.32).

В первую очередь необходимо щелкнуть по строке **Создайте резюме** (рис. 6.33).

Далее, заполнив указанные поля, следует нажать кнопку **Сохранить** (рис. 6.34), тогда резюме попадет в БД и будет доступно для просмотра работодателям.

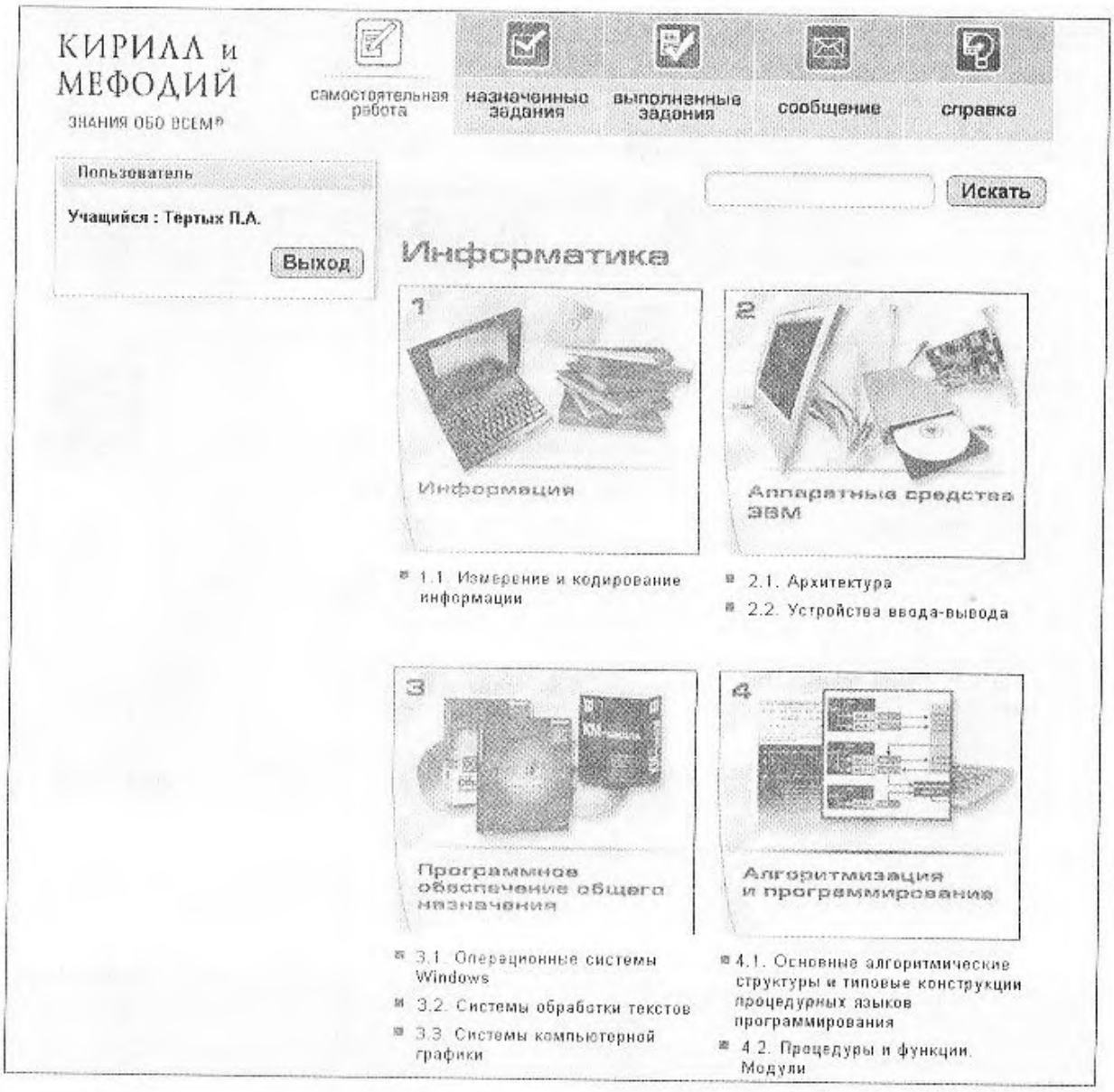


Рис. 6.27. Личная страница пользователя на сайте http://webpractice.cm.ru

Параллельно с простым размещением резюме, для того чтобы повысить эффективность поиска, можно просматривать на этом же (и подобных) сайте имеющийся банк вакансий, и если какая-то из них вам приглянется, выслать резюме на электронный адрес работодателя (рис. 6.35).

Вопросы и задания

1. Что такое телекоммуникации?
2. Что представляют собой сервисы коллективного взаимодействия пользователей Интернета?
3. Войдите на один из сайтов, посвященных поиску работы, составьте резюме по предлагаемому шаблону, осуществите поиск в банке вакансий.
4. Зарегистрируйтесь и примите участие на форуме «Информационные технологии в образовании» на сайте http://pedsovet.org/forum/.

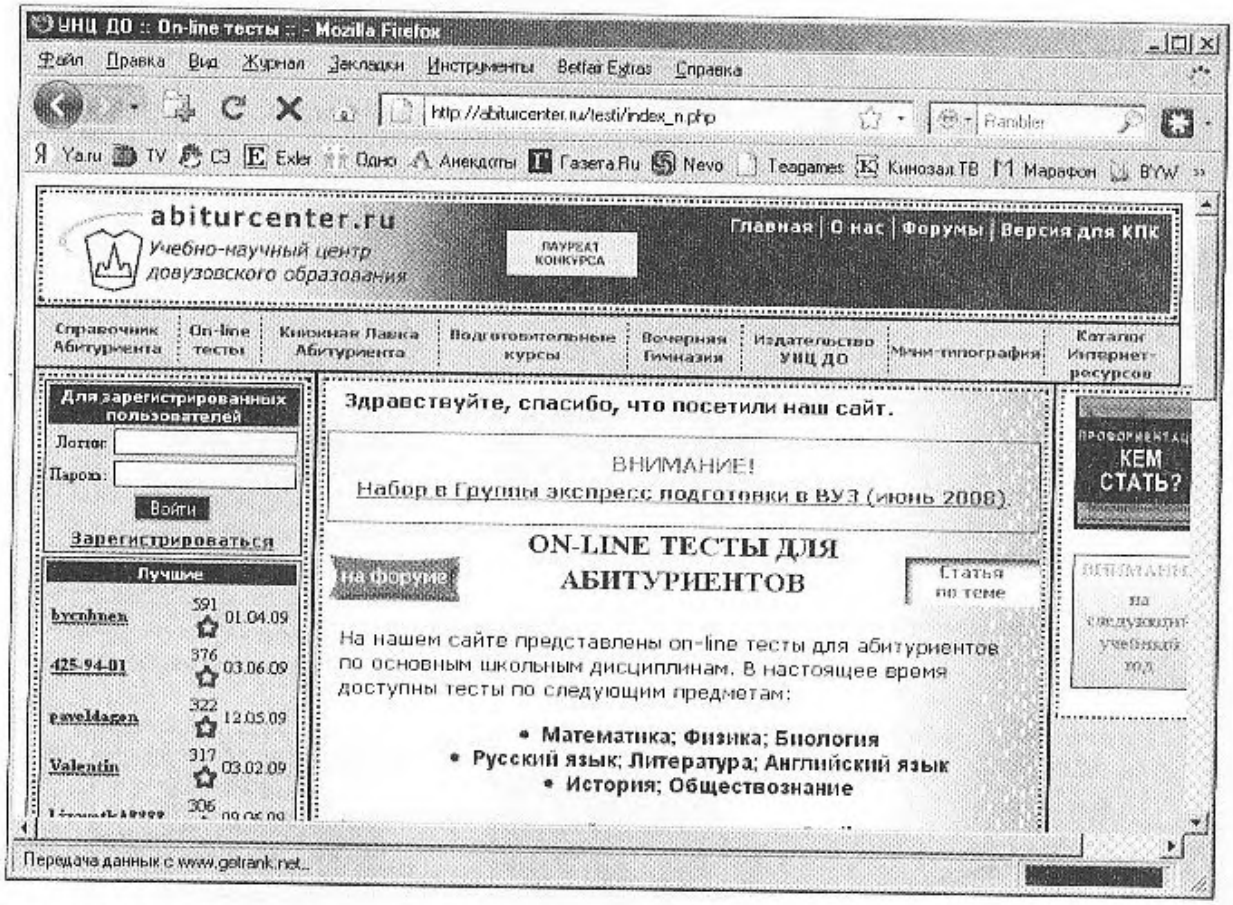


Рис. 6.28. Онлайн-тесты ЕГЭ для абитуриентов

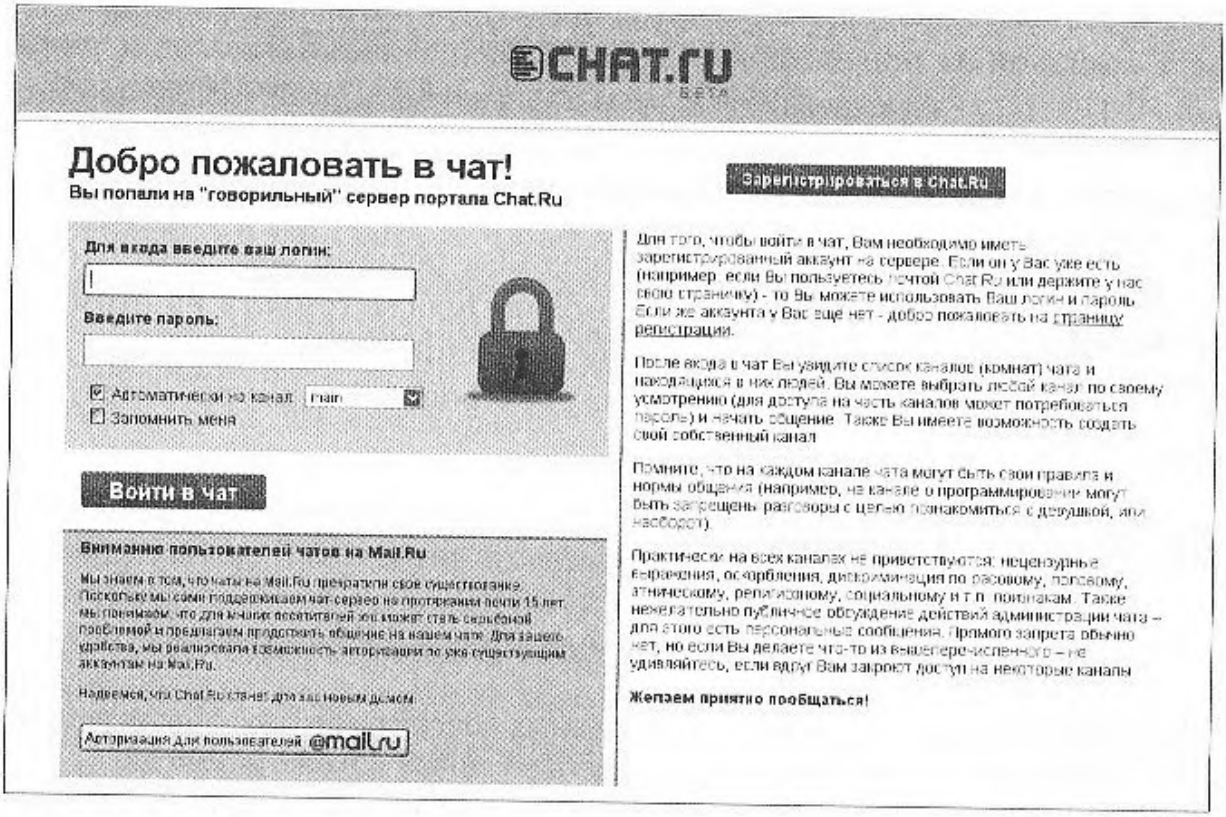


Рис. 6.29. Вход в чат mail.ru

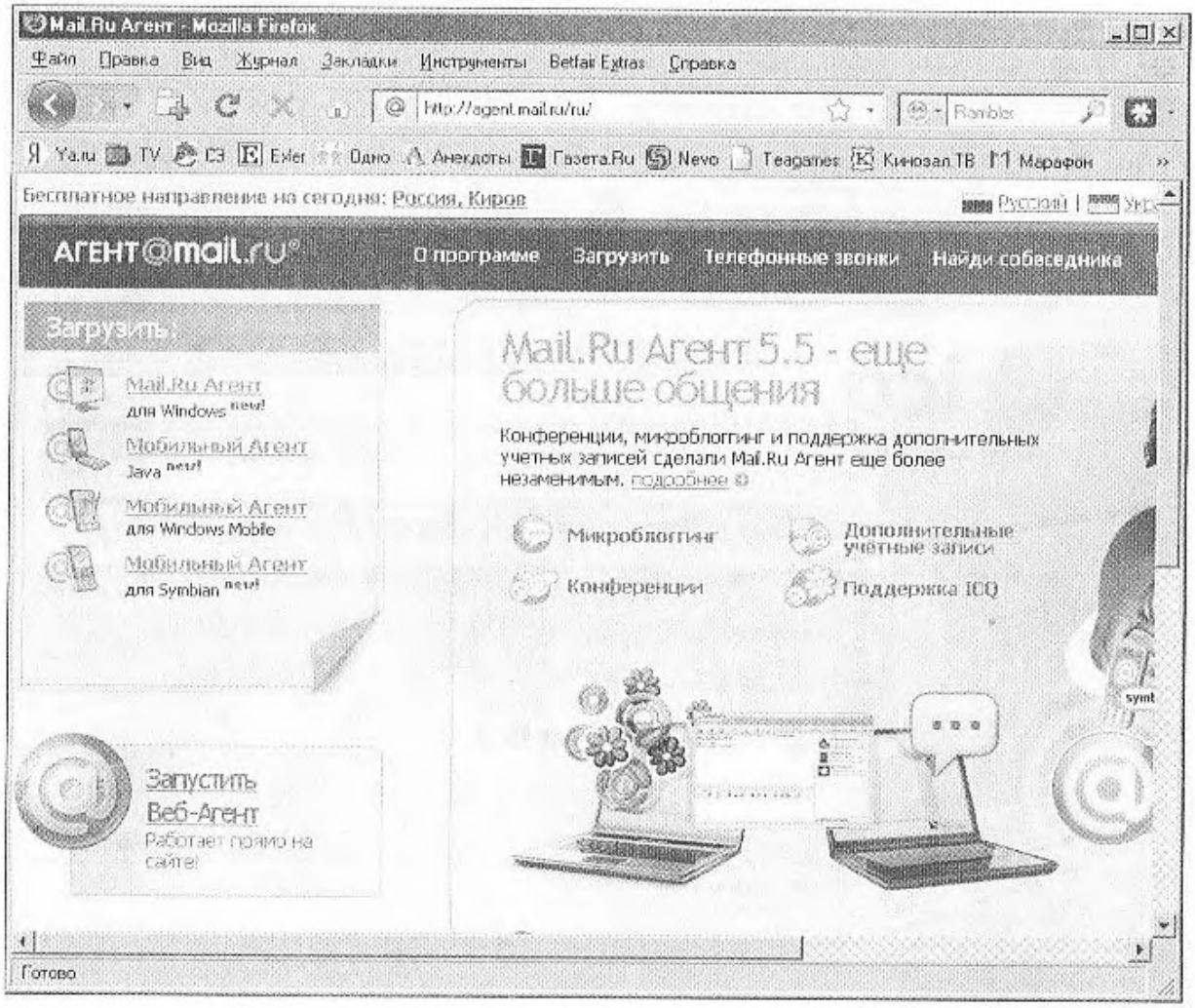


Рис. 6.30. Mail.ru Агент

5. Воспользуйтесь мобильным телефоном для получения электронной почты со своего почтового ящика.

Проектное задание

Пройдите самостоятельное тестирование на сайте сетевого практикума по информатике http://webpractice.cm.ru, используя уровень 2 и все темы курса. Оцените свои знания.

6.6. Пример работы в телеконференции на основе Skype

Дистанционные телеконференции часто проводятся с помощью программы интернет-телефонии Skype. Эта программа свободно распространяется (http://www.skype.com). Чтобы загрузить Skype, надо пройти по ссылке **Загрузить Skype** и далее выбрать устройство (компьютер) и операционную систему (рис. 6.37).



Рис. 6.31. Сайт www.icq.com

Далее нужно выбрать кнопку **Загрузить** и зарегистрироваться.

Для работы с программой необходимо подключить к компьютеру микрофон и наушники (рекомендуется использование «гарнитуры» — наушников, совмещенных с микрофоном). Для передачи видеоизображения необходима любая web-камера со значком Skype на коробке. Иногда web-камера может быть дополнена микрофоном с наушниками. Минимальные требования для web-камеры смотрите в ее инструкции.

Установка Skype на компьютере не вызывает особых сложностей и описана в имеющихся на сайте руководствах. Необходимо лишь обратить внимание на выбор имени пользователя Skype (Skype Name) при создании «нового пользователя» Skype (Skype Account): вы должны ввести в соответствующих полях свое полное имя, а также имя пользователя Skype, длиной от 6 до 32 символов, по которому другие пользователи смогут однозначно найти вас для связи через Skype. Рекомендуется также указать такие данные: e-mail, страну, город и родной язык — это может существенно облегчить другим пользователям поиск вашей контактной информации.

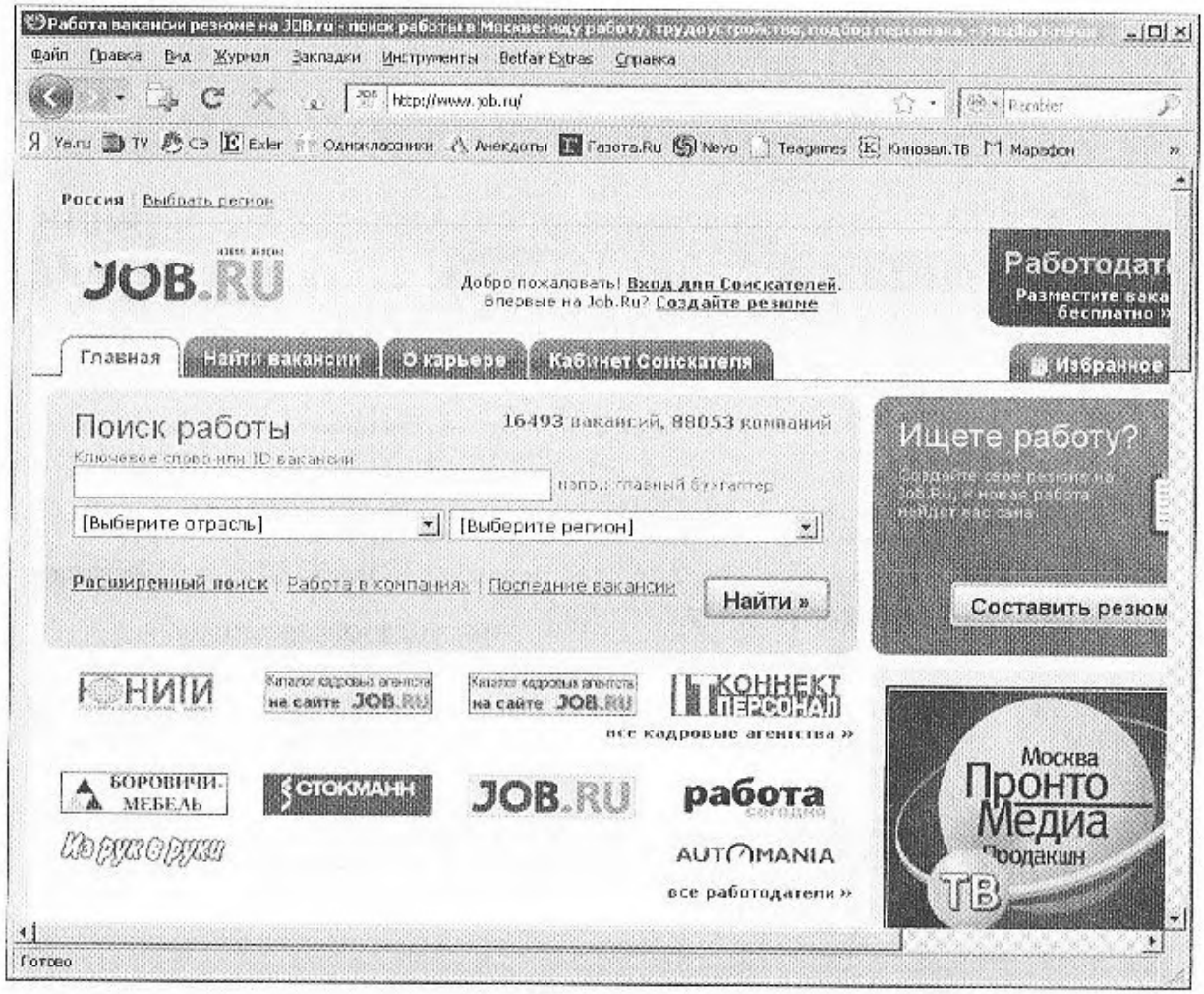


Рис. 6.32. Поиск работы на сайте Job.ru

Настройка программы Skype достаточно проста. При использовании микрофона и наушников следует настроить громкость звука и чувствительность микрофона, а затем проверить работу с программой, совершив тестовый звонок по изначально предусмотренному в Skype контакту «Тест-Звонок Skype» (автоответчик, предлагающий после сигнала записать произвольную голосовую фразу, а затем прослушать ее воспроизведение). Для совершения звонка через Skype достаточно выбрать в окне Skype вкладку **Контакты**, затем требуемого абонента (в данном случае — «Тест-Звонок Skype») и щелкнуть мышью по кнопке ; щелчок мышью по кнопке завершает звонок (рис. 6.36).

Настройка работы с цифровой видеокамерой производится несколько сложнее и зависит от конкретного типа видеокамеры.

Добавление контакта. Для связи с абонентом Skype, осуществляющим индивидуальные консультации или проводящим общую телеконференцию, нужно будет добавить его в свой список контактов. Для этого необходимо:

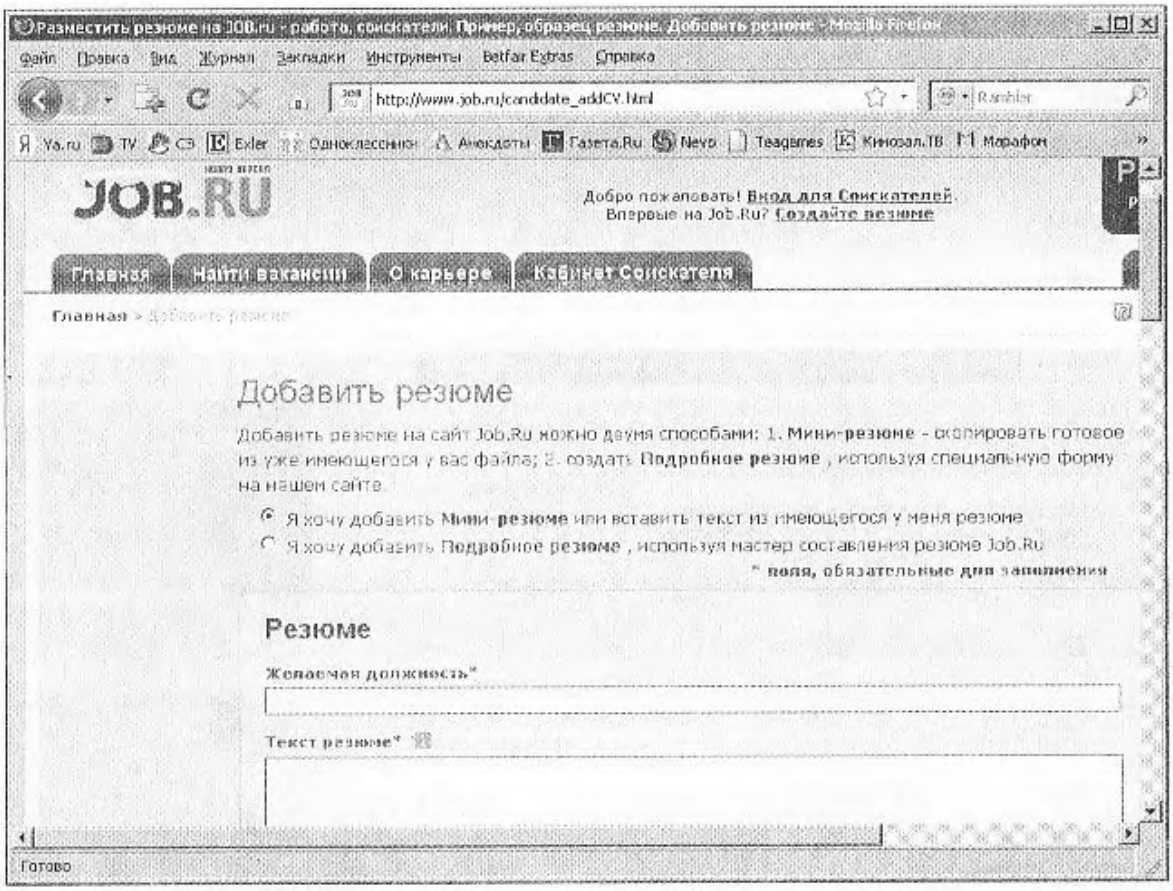


Рис. 6.33. Добавление резюме

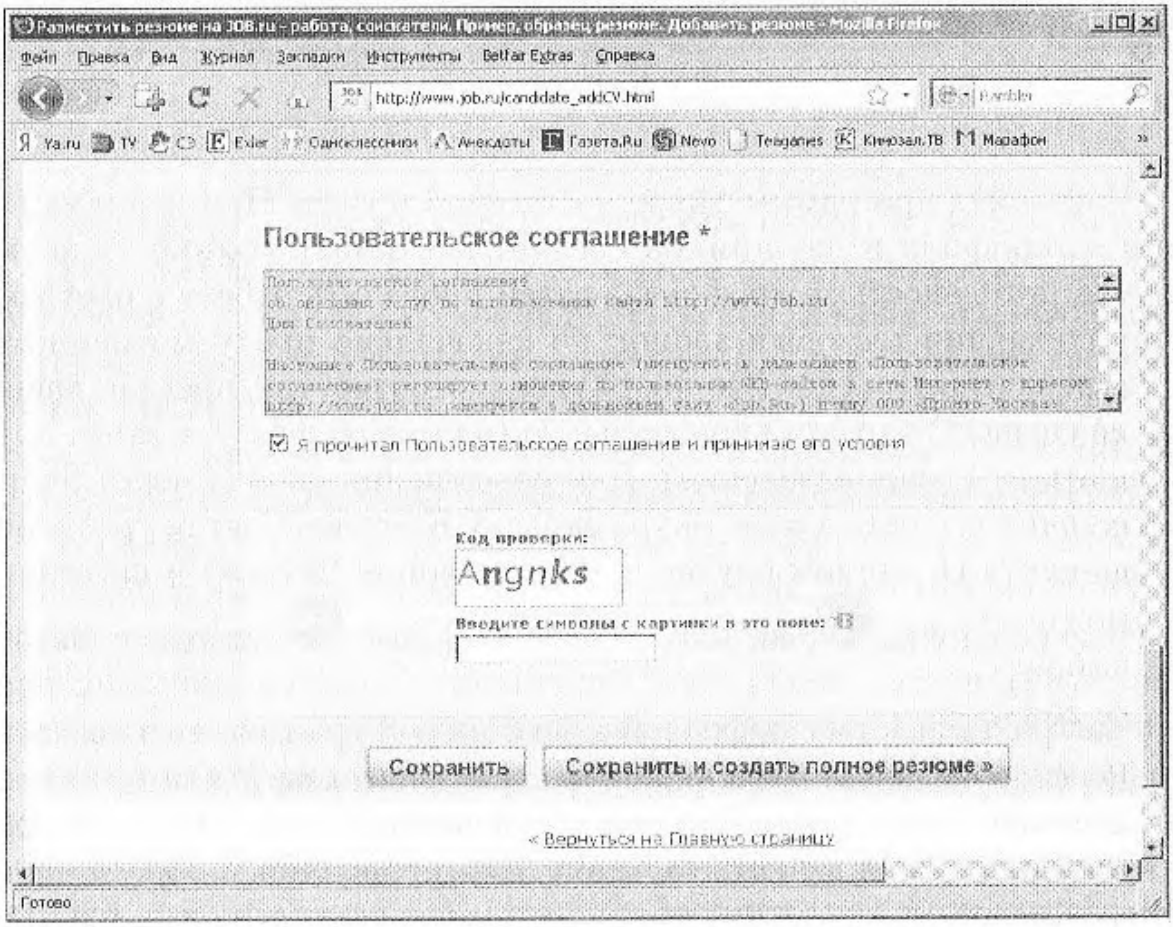


Рис. 6.34. Сохранение резюме

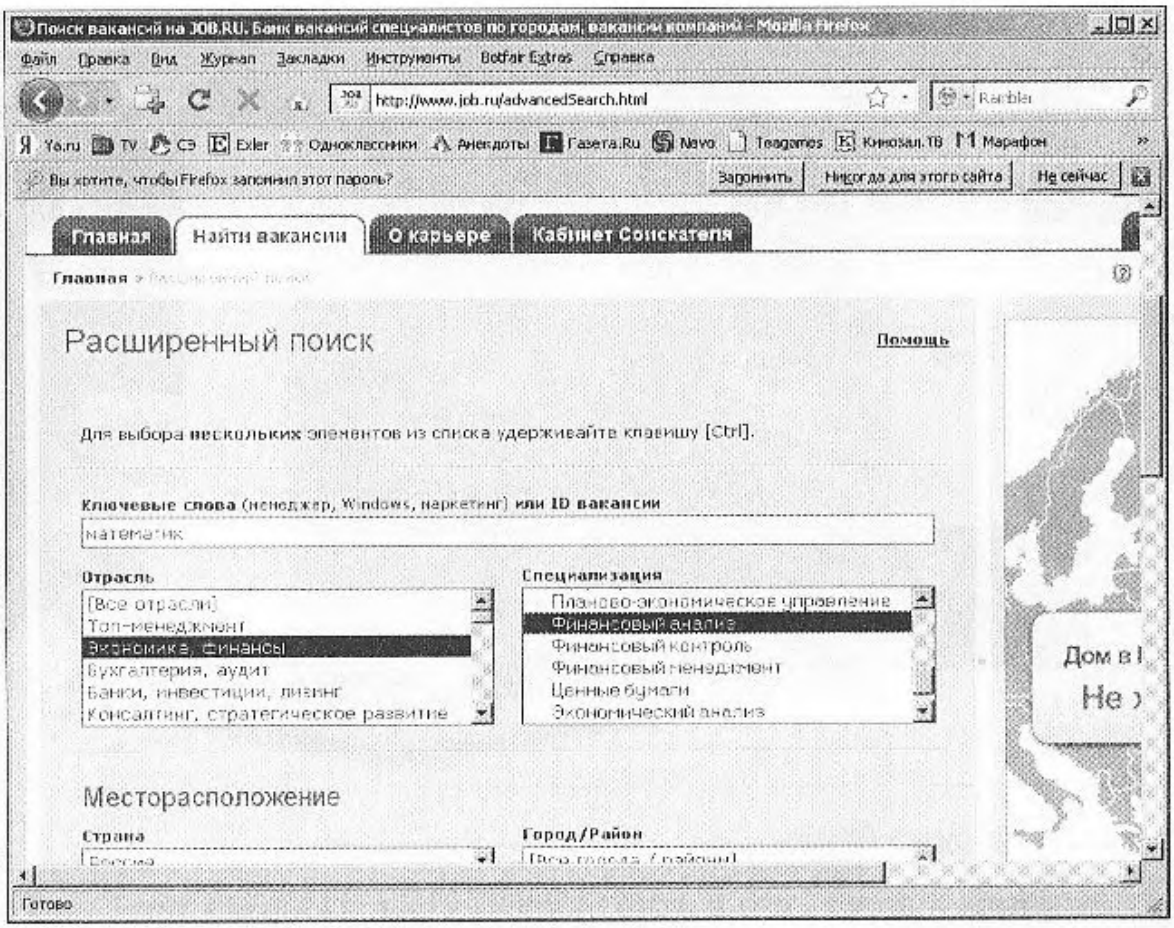


Рис. 6.35. Поиск вакансий



Рис. 6.36. Тест-Звонок Skype

- выбрать в главном меню Skype пункт **Контакты**, **Добавить новый контакт**;
- в открывшемся окне заполнить одно или более из известных вам полей, после чего автоматически начнется поиск контакта;
- щелкнуть мышью на кнопке **Посмотреть** — появится таблица возможных абонентов;
- выбрать в таблице нужный вам контакт, щелкнуть мышью по кнопке **Добавить контакт**.

В результате в окне Skype на вкладке **Контакты** появится строка с указанием имени соответствующего абонента.

Видеозвонок. Когда абонент находится на связи, значок Skype слева от его имени имеет зеленый цвет (с белым символом «галочка»). Если абонент в этот момент свободен (не ведет разговор), вы сможете об-

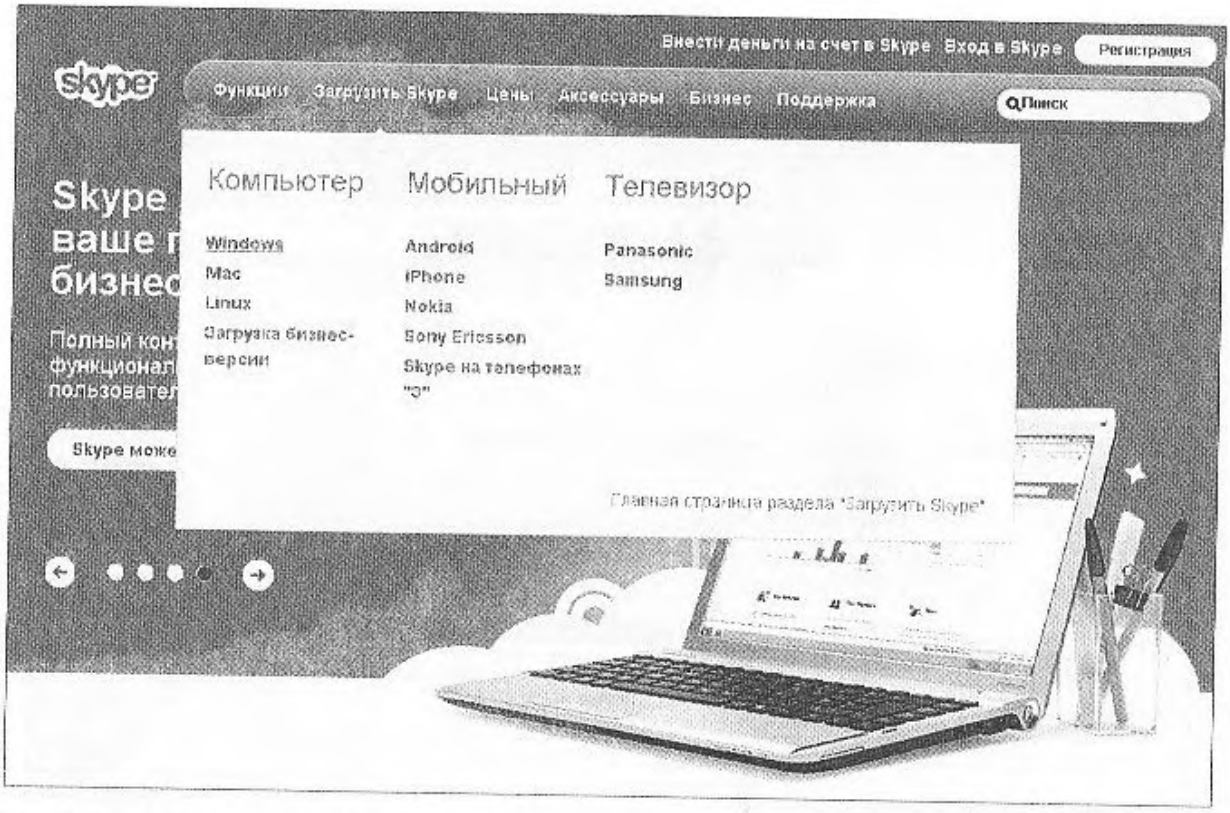


Рис. 6.37. Загрузка Skype

щаться с ним и видеть его в окне Skype в кадре изображения с видеокамеры этого абонента (рис. 6.38).

Проведение голосовых телеконференций. При организации коллективных обсуждений ведущий создает в Skype отдельную голосовую телеконференцию, к которой может пригласить несколько абонентов из своего списка контактов.

Для этого ведущему необходимо:

- в указанное время проведения телеконференции включить Skype;
- надеть гарнитуру (наушники с микрофоном);
- выбрать первого абонента и начать общение;
- войти в список контактов и телеконференций, поочередно выбрать нужного абонента и «Добавить в разговор», щелкнув по соответствующей кнопке (см. рис. 6.38).

После выполнения соединений в окне Skype будет отображаться список всех участников данной конференции (рис. 6.39).

Skype предоставляет возможность слушать выступления ведущего или других участников конференции; если есть желание попросить слова, нужно щелкнуть мышью по кнопке **Попросить слова** в верхней части списка участников — ваше имя появится в списке ожидающих, и когда до вас дойдет очередь, вы сможете начать выступление.

Прервать участие в конференции можно при помощи кнопки . В этом случае будет предложено (если вы не отключили его) окно со

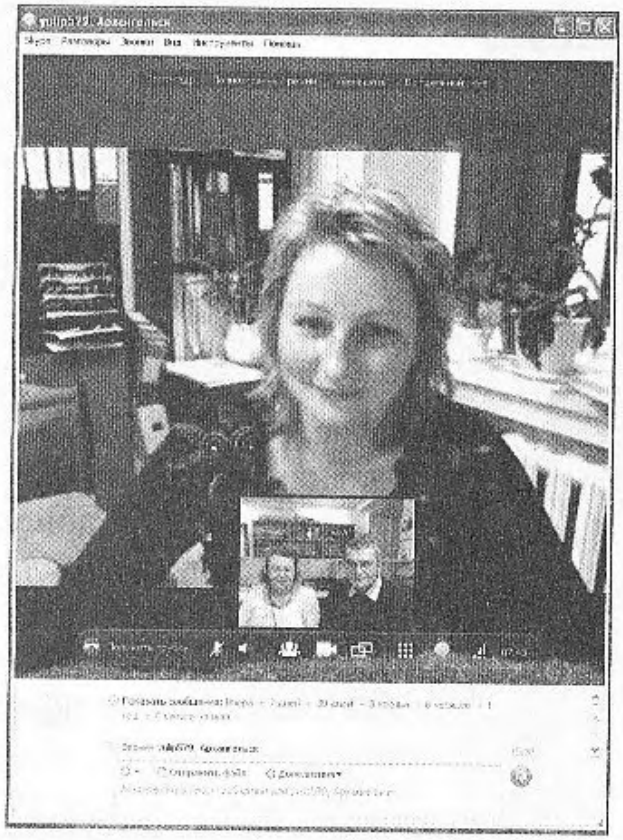


Рис. 6.38. Подключение видеозвонка

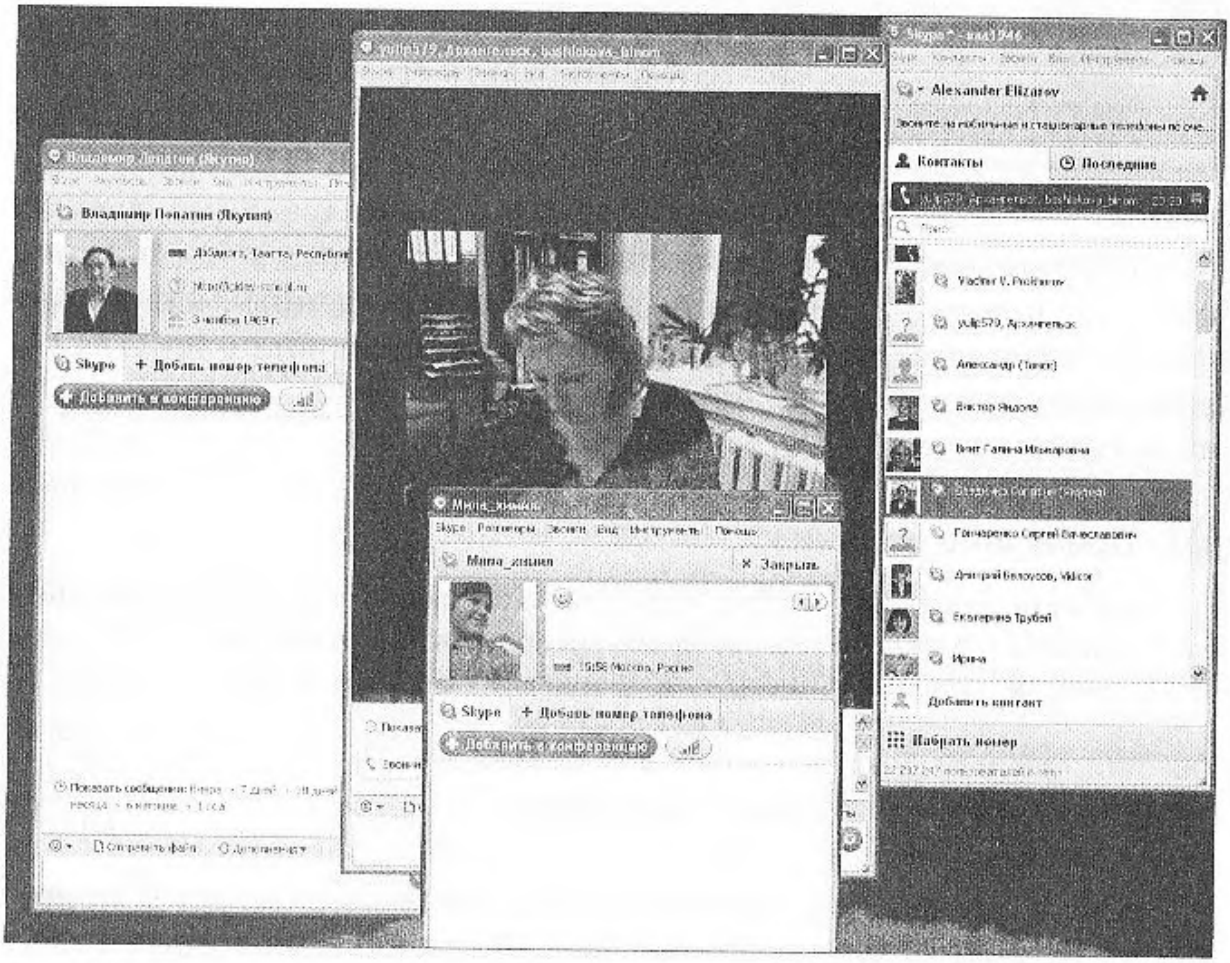


Рис. 6.39. Организация голосовой конференции

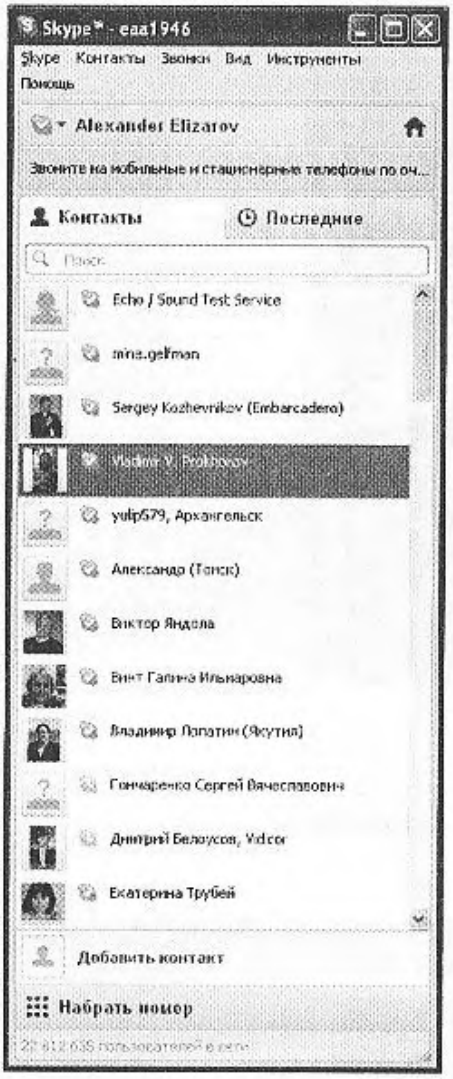


Рис. 6.40. Основное окно Skype: контакты

списком всех участников конференции, с помощью которого вы можете, пометив соответствующие флажки, добавить тех или иных участников в свой список контактов для последующего индивидуального общения через Skype (рис. 6.40).

Для возврата к списку ваших абонентов необходимо снова выбрать в Skype вкладку **Контакты**.

Чат-режим. В дополнение к голосовому общению консультант может пригласить вас присоединиться к организованному им групповому чату, либо вы можете вызвать чат-режим для письменного общения с консультантом и (или) другими участниками. Для этого следует выбрать в списке абонентов на вкладке **Контакты** желаемого абонента, щелкнуть мышью на его кнопке и в поле чата ввести текст. Затем отправить его, нажав **Enter**. Далее нужно дождаться ответа и продолжить обмен текстами (рис. 6.41).

Просматривать полученные сообщения можно в верхнем окне, а писать свои сообщения — в нижнем окне. Есть также возможность добавлять в созданный чат других абонентов, пересылать файлы другим участникам чата и пр. Для сокращения текста в чате часто используются специальные значки — «смайлики». (рис. 6.42).

Проектное задание

В режиме телеконференции, чата или голосовой конференции проведите коллективное обсуждение по проблемам использования Интернета в досуговой деятельности, используя инструкцию, описанную в параграфе.

6.7. Сетевая этика и культура

Особое значение при использовании компьютерных сетей приобретает сетевая культура, в том числе этика общения в сети. Очень многие сейчас связывают информационное общество именно с Интер-

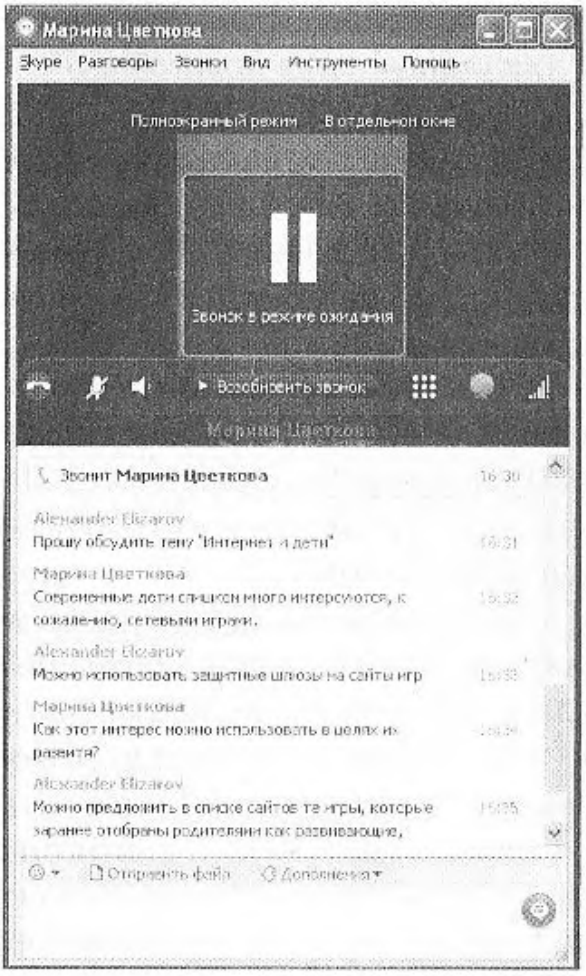


Рис. 6.41. Окно чата

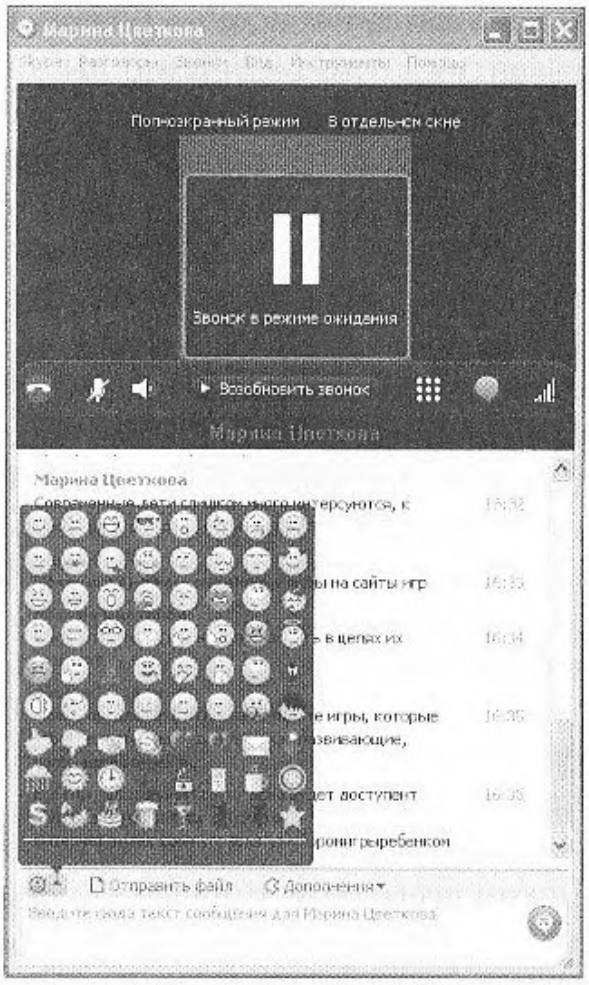


Рис. 6.42. Символы «смайлик» для чата

нетом как основой новейших информационных отношений. Интернет — это объединяющее начало, с которого, собственно, и начался переход к информационному обществу.

Учитывая неуправляемый, постоянно растущий объем самой разной информации в Интернете, этике в ней придается немаловажное значение. С каждым днем вопрос об этических нормах в Интернете приобретает все большую актуальность. Ведь для того, чтобы Интернет стал поистине средством массовой коммуникации людей, несомненно, нужно придерживаться общепринятых этических норм и ценностей.

Интернет как средство массовой коммуникации достиг такого уровня развития и воздействия на общественную жизнь, который требует государственного вмешательства в виде принятия законов, регулирующих деятельность, связанную с Интернетом, ибо государство выступает защитником своих граждан, в том числе и от некачественной или аморальной информации. Однако и простые пользователи сети не должны забывать про этические нормы. Сейчас в Интернете существует достаточно много сайтов, блогов, чатов и конференций, посвященных данному вопросу. По мнению большинства исследова-

телей, данная проблема очень важна как для более упорядоченного развития Всемирной паутины, так и для становления нового высокоморального человека — члена информационного общества.

В целях определения направления такого регулирования в ряде стран были проведены специальные исследования, созданы общественные объединения, разрабатываются кодексы поведения во Всемирной сети, принимаются законы. Все упомянутые документы доступны в Интернете, что делает излишним их пересказ или подробный анализ. Вместе с тем краткая их характеристика дает возможность понять направления развития сетевого законодательства, этики и опыта регулирования Интернета. Интернациональная природа Интернета позволяет утверждать, что проблемы законодательного регулирования деятельности в Сети носят общий характер и одинаковы для всех стран. И именно озабоченность государства перед перспективой иметь рядом с собой такой неконтролируемый организм, как Интернет, способный влиять на сознание граждан, побуждает активно мыслящее мировое сообщество к правовому регулированию сетевых отношений, в том числе созданию и поддержке *сетевой этики*.

Интернет становится тотальным в планетарных масштабах источником всевозможнейшей информации и средой интерактивного общения. С развитием Интернета в обиход входит понятие *нэтикета* (от англ. *net* — сеть и *etiquette* — этикет), обозначающее совокупность правил, сложившихся среди пользователей глобальной сети. Первый *Кодекс компьютерной этики* был разработан в 1979 г. в США. Принятие кодекса было продиктовано пониманием того, что инженеры, ученые и технологи результатами своей деятельности определяют качество и условия жизни всех людей в информационном обществе.

Принципами, разработанными в рамках компьютерной этики, являются:

- *тайна частной жизни* (*privacy*) — право человека на автономию и свободу в частной жизни, право на защиту от вторжения в нее органов власти и других людей;
- *точность* (*accuracy*) — соблюдение норм, связанных с точным выполнением инструкций по эксплуатации систем и обработке информации, честным и социально-ответственным отношением к своим обязанностям;
- *частная собственность* (*property*) — неприкосновенность частной собственности — основа имущественного порядка в экономике. Следование этому принципу означает соблюдение права собственности на информацию и норм авторского права;
- *доступность* (*accessibility*) — право граждан на информацию, ее доступность в любое время и в любом месте.

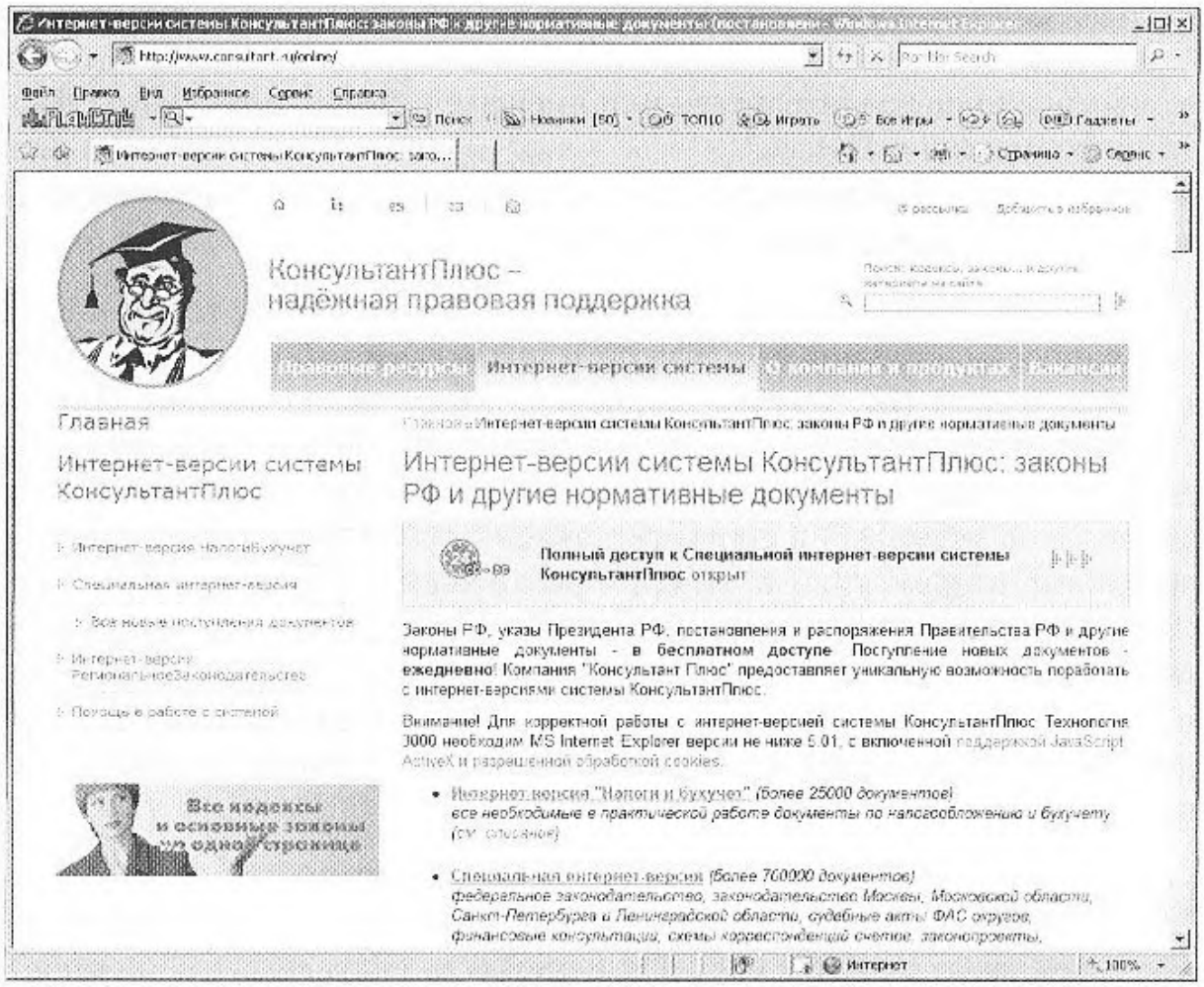


Рис. 6.43. Интернет-версия системы «Консультант Плюс»

В обычном обществе каждый может претендовать на индивидуализм, т. е. личную свободу, но во многих случаях индивидуализм должен идти на компромисс с интересами большой социальной группы, быть законопослушным. Никто не станет полагать, что Сеть живет вообще без проблем и злоупотреблений. Однако если эти проблемы не разрешаются самим сетевым сообществом, они могут стать объектом правовой защиты и даже судебного разбирательства. Важно подчеркнуть, что принцип доступности государственной информации по Интернету принят многими странами как одно из условий развития демократических прав граждан.

Стоит перечислить некоторые действия, считающиеся в Интернете «вредными» привычками или правонарушениями, которых, соответственно, следует избегать:

- чрезмерное увлечение пользователя сети играми во вред обучению или профессиональной деятельности;
- чрезмерное пребывание в Интернете, нарушающее нормальный ритм жизни;

- агрессивное, надоедливое и иное антиобщественное поведение в Сети;
- умышленное вредительство или вмешательство в дела других, в их информацию и ПО;
- публикация файлов агрессивного или аморального содержания;
- нарушение авторских прав, плагиат.

Отыскать *правовые акты и законы* по перечисленным темам и другим вопросам, связанным с этикой и правилами работы в Интернете в юридической БД, можно, например, в базе «Консультант Плюс» (http://www.consultant.ru/online/) (рис. 6.43). Эта база данных содержит правовую и нормативную информацию для россиян. Она удобна тем, что в ней в соответствующих рубриках собраны законодательные и нормативные акты по различным тематическим направлениям, а также тем, что информация постоянно актуализируется (обновляется). Эта база необходима для работы руководителей, экономистов, юристов, ученых, студентов, а также всех тех, кому необходим доступ к выверенной и актуальной информации нормативно-правового характера.

? Вопросы и задания

1. В чем заключается сетевая этика и сетевая культура?
2. Перечислите принципы поведения в сети. Охарактеризуйте каждый принцип.
3. Каковы основные правила сетевой этики?

П-СЭ Проектное задание

Используя базу «Консультант плюс» (http://www.consultant.ru/online/), найдите правовые нормы, реализующие ответственность граждан за нарушение правил использования лицензионного ПО, регулирования авторских прав на электронные источники информации, на размещение в сети информации аморального или агрессивного содержания.

ЗАКЛЮЧЕНИЕ

Курс информатики, представленный в данном учебнике, охватил шесть основных тем: информационная деятельность человека, информация и информационные процессы, средства информационных и коммуникационных технологий, технологии создания и преобразования информационных объектов и технологии работы с информационными структурами — электронными таблицами и базами данных, телекоммуникационные технологии.

Предлагаем вместе еще раз проследить и обдумать основные понятия курса, ставшие частью вашей информационной культуры.

Глава 1. Информационная деятельность человека:

- информационное общество, средства массовой коммуникации, средства связи, ЭВМ, микропроцессорная техника, «киберпространство», индустриальное общество и постиндустриальное общество;
- информационная деятельность человека, робототехника, САПР — система автоматизированного проектирования, АСУ — автоматизированная система управления, дистанционное обучение, компьютерная диагностика, бортовой компьютер, цифровое оборудование;
- персональный компьютер, информационные ресурсы, цифровые образовательные ресурсы, образовательный портал, «единое окно», электронные книги, гиперссылки, электронные библиотеки.

Глава 2. Информация и информационные процессы:

- информатика, информация, передача информации, источник информации, свойства информации, энтропия — мера неопределенности, мера информации, кибернетика, информационно-управляющие закономерности, системы управления (размокнутые, самоуправляемые или замкнутые), прямая связь и обратная связь;
- модель, моделирование, информационные признаки моделей, аналогово-цифровое и цифроаналоговое преобразование, дискрет, модуляция и демодуляция, информационные модели, алгоритмизация, оптимизация, структурные информационные модели, базы данных, системы управления базами данных (СУБД), табличная,

сетевая и иерархическая модель данных, формальная модель, параметры модели, компьютерная модель;

- количество информации, единицы измерения (бит, байт, килобайт), системы счисления (позиционные и непозиционные), основание системы счисления, перевод чисел из одной системы счисления в другую, арифметические операции, двоичные разряды, тетрада, трада, триггер, регистр, интегральная схема;
- символьная информация (символьные коды, символы кириллицы), звуковая информация (частота звучания, длительности звучания, звукозапись), графическая информация (растровая графика, пиксел, кодировка цвета и координаты пикселов), разрешающая способность монитора, разрешение изображения, векторная графика (характеристики линии, аналитическая или вычисляемая графика), масштабирование, цветовое разрешение, цветовые модели;
- хранение информации, файл (чтение, запись, исполняемые файлы, текстовые файлы, имя файла, расширение, атрибуты), магнитный диск (фрагменты, таблица размещения файлов, треки, объем памяти, кластер, сектор), каталог (иерахическая система, директория, корневой или главный каталог, подкаталоги), доступ к информации, файловая система (цепочки FAT, метафайлы, потоки);
- обработка информации, алгоритм (шаг, свойства алгоритмов, линейная алгоритмическая конструкция, разветвляющаяся конструкция, условие, логическое выражение, алгебра логики, циклическая конструкция, счетчик, вложенные циклы, индекс, массив), алгоритм перебора, «пузырьковая» сортировка, структурная схема алгоритма;
- алгоритмический язык и компьютерная программа, системы программирования (язык программирования, транслятор, библиотека программ), технология программирования, проблемно-ориентированное программирование, универсальный язык программирования, процедурно-ориентированное программирование, отладка, принцип структурного программирования, техника модульного программирования, объектно-ориентированное программирование, логическое программирование, визуализация программирования, тестирование программ;
- синтаксис языка программирования (алфавит языка, зарезервированные слова, слова пользователя, идентификаторы, типы данных, встроенные функции, операторы языка программирования), трансляция программы, семантика программы (оператор присваивания, арифметическое выражение, оператор ввода и вывода, оператор безусловного перехода, оператор условного перехода, оператор цикла, тело цикла, счетчик цикла).

Глава 3. Средства информационных и коммуникационных технологий:

- перфокарты и код Холлерита, управление командами и машинно-счетные станции, ЭВМ первого поколения и электронные лампы, ЭВМ второго поколения и транзистор, ЭВМ третьего поколения и микросхемы, семейства ЭВМ, операционные системы, ЭВМ четвертого поколения и большие интегральные схемы;
- персональный компьютер, архитектура компьютера, микропроцессор, системная шина, основная память, внешняя память, периферия, средства мультимедиа, магистрально-модульный принцип, открытая архитектура, принцип программного управления компьютером;
- алгебра высказываний, константы алгебры логики, логическое умножение, логическое сложение, логическое отрицание, логические схемы, логическое выражение, таблицы истинности, элементная база компьютера, сумматор, вентильные схемы, логические диаграммы;
- программное обеспечение, системное программное обеспечение, прикладное программное обеспечение, графический интерфейс, рабочий стол, ярлык, файловый менеджер, драйверы, «свободное» программное обеспечение, дистрибутив, архиватор, антивирусные программы и антивирусная защита.

Глава 4. Технологии создания и преобразования информационных объектов:

- интегрированные среды обработки информации, текстовый редактор, текстовый процессор, абзац, форматирование, настройка страницы, верстка, панель инструментов, шрифт, таблица и границы, иллюстрации в тексте, шаблон, гипертекст, оглавление;
- графический редактор, меню, отмена ошибочного действия, сканирование, видеомонтаж, раскадровка, видеоэффекты, чертеж;
- звуковой редактор, студия звукозаписи, звуковые колонки, звуковая карта, MIDI-кабель, мультимедиа, фонограф, универсальный проигрыватель, микшер;
- компьютерная презентация, мастер презентаций, слайд.

Глава 5. Технологии работы с информационными структурами — электронными таблицами и базами данных:

- компьютерный вычислитель, инженерный калькулятор, ЭТ — электронные таблицы (столбец, строка, диапазон, ячейка, адрес, формат ячеек, защищенные ячейки), Мастер диаграмм;
- реляционная модель данных, домен, отношения, атрибут отношения, кортеж отношения, мощность отношения, поиск данных в таблице, параметры поиска, поиск по условиям, поиск по фильтрам, БД —

база данных, СУБД — система управления базой данных, поле базы данных, запись в базе данных, ключевое поле (ключ), поиск с помощью запросов, сортировка, отчеты.

Глава 6. Телекоммуникационные технологии:

- компьютерные сети, обмен данными, распределенный доступ (логин, пароль), сервер, маршрутизатор, модем, протоколы (межсетевой протокол IP и транспортный протокол TCP), Word Wide Web — Всемирная паутина, протокол http, браузер, web-сайты, система адресации к web-документам, имя домена;
- ЛВС — локальная компьютерная (вычислительная) сеть, одноранговые и двухранговые сети, сетевая операционная система, рабочая станция, сетевая плата, сервер, кабели и устройства беспроводной связи, концентратор или хаб, топология локальной сети («шина», «звезда», «кольцо»), аппаратный адрес, символьные адреса;
- браузер, web-страница, web-технология, язык HTML, теги, web-редакторы, конструкторы сайтов;
- личные сетевые сервисы, персонализация, регистрация, электронная почта, почтовый ящик, почтовый сервер, файлообмен, поисковый сервер (новости, репортажи, телепередачи, интернет-СМИ), web-камера, интернет-репортаж, электронная доска объявлений, конференции, дистанционное обучение, онлайн-тестирование;
- сетевые услуги коллективного использования, коллективное обсуждение, чат, служба мгновенных сообщений, форум, интернет-телефония Skype;
- сетевая культура, сетевая этика, кодекс компьютерной этики, правовые акты и законы информационной безопасности.

Вы можете использовать приведенный справочник логического развития основных понятий для каждой темы в качестве самоконтроля и самооценки их освоения. Если вы логически связываете предложенные наборы понятий в единый рассказ, можно с уверенностью сказать, что задача встраивания информатики в вашу информационную культуру решена. А вот свою активность в информационной деятельности вы можете оценить по результатам выполнения проектных заданий, выбранным вами в соответствии с профилем обучения.

Надеемся, что результатом освоения курса для вас стало формирование общей информационной культуры и готовности к активной информационной деятельности, а также приобретение навыков саморазвития и самообразования с помощью новых информационных технологий в сфере ваших профильных интересов.

ПРИЛОЖЕНИЕ

Список открытых интернет-ресурсов

Название	Ссылка
Государственный образовательный портал. Раздел «Абитуриент»	http://edu.ru
Коллекции цифровых образовательных ресурсов. Раздел «Информатика и ИКТ», 8—11 классы, коллекция Политехнического музея	http://school-collection.edu.ru
Каталог Федерального центра информационно-образовательных ресурсов. Раздел «Среднее (полное) общее образование / Информатика и ИКТ». Разделы НПО и СПО	http://fcior.edu.ru
Портал «Информационно-коммуникационные технологии в образовании»	http://www.ict.edu.ru
Открытый сетевой компьютерный практикум по курсу «Информатика и ИКТ» компании «Кирилл и Мефодий»	http://webpractice.cm.ru
Блоги НПО на сайте «Педсовет»	http://pedsovet.org/
Справочник профессий	http://www.rabota-in.net/prof/professions.php

Продолжение таблицы

Название	Ссылка
Специальности СПО	http://www.edu.ru/abitur/act.11/index.php, http://www.job.ru, http://www.superjob.ru, http://www.rabota.ru
Информационный ресурсный центр по практической психологии (тесты по профориентации)	http://psyfactor.org/tests/
Модели биоритмов (онлайн)	http://www.ritms.ru/, http://ru.wikipedia.org/wiki/Биоритм
Среда программирования QBASIC	http://www.softportal.com/get-2408-microsoft-quickbasic-qbasic.html),
Среда программирования Free Pascal	http://www.freepascal.org/ download.var, http://www.scintilla.org/
Файловый менеджер Frigate	http://www.frigate3.com/download/frigate3_pro.exe
Операционные системы (для серверов и рабочих станций)	http://linux.armd.ru, http://ubuntu.ru/, http://www.debian.org/,
Офисный пакет (текстовый редактор Writer, редактор математических формул Math, электронная таблица Calc, система управления базами данных Base, редактор презентаций Impress	http://ru.openoffice.org/
Визуальная среда программирования	http://scratch.mit.edu/, http://www.lazarus.freepascal.org/, http://www.eclipse.org/, http://www.netbeans.org/
Web-браузер	http://www.mozilla-europe.org/ru/firefox/ http://www.google.com/chromo/ http://www.opera.com/download/ http://www.apple.com/ru/safari/download/
Клиент электронной почты	http://www.mozilla-europe.org/ru/products/thunderbird/

Продолжение таблицы

Название	Ссылка
HTML-редактор	http://www.kompozer.net/
Редактор растровой графики	http://www.gimp.org/
Редактор векторной графики	http://www.inkscape.org/
Профессиональная система черчения	http://www.qcad.org/
Настольная издательская система	http://www.scribus.net/
Редактор звуковых файлов	http://audacity.sourceforge.net/
Видеоредактор	http://www.avidemux.org/
Редактор трехмерной графики	http://www.blender.org/
Видеосвязь, служба мгновенных сообщений	http://www.skype.com
Software as a Service (ПО как сервис)	http://www.google.com/a/help/intl/ru/index.html
Антивирусные программы	http://www.bestfree.ru, http://fantivirus.ru/, http://www.antivirus-navigator.com/free_antiviruses.htm
Клавиатурный тренажер. Раздел «Информатика», 7–9 классы, тренажер «Руки солиста»	http://www.school-collection.edu.ru
Переводчики онлайн	http://www.translate.ru/, http://translate.google.com/#
Графический редактор VISIO — триальная версия	http://technet.microsoft.com/ru-ru/officie/ee236295
Система черчения «КЕДР»	http://www.kedrin.com/download.htm

Окончание таблицы

Название	Ссылка
Автоматизированная система проектирования ArchiCAD	http://www.archicad.ru/company/buy.html
АСУ «Парус» (демоверсия)	http://www.oknemika.ru/body1100.asp
Сайт компании CISCO, учебный сайт проектирования ЛВС учреждения	http://cisco.novsu.ru/
Web-редактор Visual Workshop	http://www.snkey.net/download/products/
Поисковые системы	www.yandex.ru, www.google.ru, www.yahoo.com, www.aport.ru
Сетевые сервисы общения	http://chat..chat..ru/ http://www.icq.com, http://www.skype.com
Юридическая база данных «Консультант плюс»	http://www.consultant.ru/online/
Энциклопедия «Кирилл и Мефодий»	www.megabook.ru
Энциклопедия «Википедия»	http://ru.wikipedia.org/wiki/
Портал Microsoft для образовательных учреждений «Школам России» (для поддержки лицензионного ПО)	http://www.shkolaedu.ru/
Сайт технической поддержки ПО Microsoft	http://technet.microsoft.com/

СПИСОК ЛИТЕРАТУРЫ

В помощь профильному изучению курса предлагается использовать дополнительные элективные курсы по информатике, а также книги по алгоритмизации и программированию:

1. Информатика и ИКТ. Базовый уровень : учебник для 10—11 классов / И. Г. Семакин [и др.]. — 4-е изд., испр. — М. : БИНОМ. Лаборатория знаний, 2008. — 246 с.

2. Информатика и ИКТ. Базовый уровень : практикум для 10—11 классов / И. Г. Семакин [и др.] . — 3-е изд., испр. — М. : БИНОМ. Лаборатория знаний, 2008. — 120 с.

3. Информатика. Задачник-практикум — В 2 ч. / И. Г. Семакин [и др.]. — Ч. 1. — 5-е изд. — М. : БИНОМ. Лаборатория знаний, 2006. — 309 с.

4. Информатика. Задачник-практикум — В 2 ч. / И. Г. Семакин [и др.]. — Ч. 2. — 5-е изд. — М. : БИНОМ. Лаборатория знаний, 2006. — 294 с.

5. *Радченко Н. Н.* Школьная информатика. Впереди экзамены : методич. пособие / Н. П. Радченко. — М. : БИНОМ. Лаборатория знаний, 2008. — 475 с.

6. *Самылкина Н. Н.* Материалы для подготовки к экзамену по информатике / Н. Н. Самылкина, И. А. Калинин, Е. М. Островская. — М. : БИНОМ. Лаборатория знаний, 2006. — 372 с.

7. *Угринович Н. Д.* Информатика и ИКТ. Базовый уровень : учебник для 10 класса / Н. Д. Угринович. — 4-е изд. — М. : БИНОМ. Лаборатория знаний, 2008. — 212 с.

8. *Угринович Н. Д.* Информатика и ИКТ. Базовый уровень : учебник для 11 класса / Н. Д. Угринович. — М. : БИНОМ. Лаборатория знаний, 2008. — 188 с.

9. *Угринович Н. Д.* Практикум по информационным технологиям / Н. Д. Угринович, Л. Л. Босова, Н. И. Михайлова. — 6-е изд. — М. : БИНОМ. Лаборатория знаний, 2007. — 394 с.

10. *Угринович Н. Д.* Преподавание курса «Информатика и ИКТ» в основной и старшей школе : методическое пособие ЭОР на компакт-дисках (3 диска) / Н. Д. Угринович. — М. : БИНОМ. Лаборатория знаний, 2008.

11. *Угринович Н. Д.* Информатика и ИКТ. Профильный уровень : учебник для 10 класса / Н. Д. Угринович. — 3-е изд. — М. : БИНОМ. Лаборатория знаний, 2008. — 387 с.

12. *Угринович Н. Д.* Информатика и ИКТ. Профильный уровень : учебник для 11 класса / Н. Д. Угринович. — М. : БИНОМ. Лаборатория знаний, 2008. — 200 с.

13. *Усенков Д.Ю.* Уроки WEB-мастера : учеб. пособие и ЭОР на компакт-диске / Д.Ю.Усенков. — М. : БИНОМ. Лаборатория знаний, 2004. — 507 с.

Элективные курсы по информатике и ИКТ

1. *Андреева Е.В.* Математические основы информатики. Элективный курс : учеб. пособие / Е.В.Андреева, Л.Л.Босова, И.Н.Фалина. — 2-е изд., испр. — М. : БИНОМ. Лаборатория знаний, 2007. — 328 с.

2. *Андреева Е.В.* Математические основы информатики : методич. пособие и ЭОР на компакт-диске / Е.В.Андреева, Л.Л.Босова, И.Н.Фалина. — М. : БИНОМ. Лаборатория знаний, 2007. — 312 с.

3. *Залогова Л.А.* Компьютерная графика. Элективный курс : учеб. пособие / Л.А.Залогова. — М. : БИНОМ. Лаборатория знаний, 2005. — 212 с.

4. *Залогова Л.А.* Компьютерная графика. Элективный курс : практикум / Л.А.Залогова. — 2-е изд. — М. : БИНОМ. Лаборатория знаний, 2007. — 245 с.

5. *Колбин Р.В.* Глобальные и локальные сети : создание, настройка, использование. Элективный курс : учеб. пособие и ЭОР на компакт-диске / Р.В.Колбин. — М. : БИНОМ. Лаборатория знаний, 2007. — 221 с.

6. *Колбин Р.В.* Глобальные и локальные сети : создание, настройка, использование. Элективный курс : методич. пособие для учителя / Р.В.Колбин. — М. : БИНОМ. Лаборатория знаний, 2008. — 55 с.

7. *Монахов М.Ю.* Создаем школьный сайт в Интернете. Элективный курс : учеб. пособие и ЭОР на компакт-диске / М.Ю.Монахов, А.А.Воронин. — М. : БИНОМ. Лаборатория знаний, 2006. — 128 с.

8. *Монахов М.Ю.* Учимся проектировать на компьютере. Элективный курс : практикум и ЭОР на компакт-диске / М.Ю.Монахов, С.Л.Солодов, Г.Е.Монахова. — 2-е изд., испр. — М. : БИНОМ. Лаборатория знаний, 2006. — 172 с.

9. *Самылкина Н.Н.* Готовимся к ЕГЭ по информатике. Элективный курс : учеб. пособие / Н.Н.Самылкина, С.В.Русаков, А.П.Шестаков, С.В.Баданина. — М. : БИНОМ. Лаборатория знаний, 2008. — 298 с.

10. *Семакин И.Г.* Информационные системы и модели. Элективный курс : учеб. пособие / И.Г.Семакин, Е.К.Хеннер. — 2-е изд. — М. : БИНОМ. Лаборатория знаний, 2007. — 303 с.

11. *Семакин И.Г.* Информационные системы и модели. Элективный курс : практикум / И.Г.Семакин, Е.К.Хеннер. — М. : БИНОМ. Лаборатория знаний, 2006. — 87 с.

12. *Семакин И.Г.* Информационные системы и модели. Элективный курс : методич. пособие / И.Г.Семакин, Е.К.Хеннер. — М. : БИНОМ. Лаборатория знаний, 2006. — 71 с.

13. *Танова Э.В.* Введение в криптографию : как защитить свое письмо от любопытных. Элективный курс : учеб. пособие / Э.В.Танова. — М. : БИНОМ. Лаборатория знаний, 2007. — 79 с.

14. *Танова Э.В.* Введение в криптографию : как защитить свое письмо от любопытных. Элективный курс : методич. пособие для учителя / Э.В.Танова. — М. : БИНОМ. Лаборатория знаний, 2007. — 47 с.

15. *Угринович Н.Д.* Исследование информационных моделей. Элективный курс : учеб. пособие и ЭОР на компакт-диске / Н.Д. Угринович. — 2-е изд., испр. — М. : БИНОМ. Лаборатория знаний, 2006. — 200 с.

Элективные курсы Microsoft серии «Партнерство в образовании» (профильный уровень, 10—11 классы)

1. Основы компьютерных сетей : учеб. пособие. — 3-е изд., испр. и доп. — М. : БИНОМ. Лаборатория знаний, 2007. — 160 с.

2. Основы компьютерных сетей : методич. пособие для учителя и ЭОР на компакт-диске. — 3-е изд. — М. : БИНОМ. Лаборатория знаний, 2007. — 55 с.

3. Основы программирования на примере Visual Basic ®. NET : учеб. пособие. — 3-е изд., испр. — М. : БИНОМ. Лаборатория знаний, 2007. — 216 с.

4. Основы программирования на примере Visual Basic ®. NETE : методич. пособие для учителя и ЭОР на компакт-диске. — 3-е изд. — М. : БИНОМ. Лаборатория знаний, 2007. — 98 с.

5. Персональный компьютер: настройка и техническая поддержка : учеб. пособие. — 3-е изд., испр. — М. : БИНОМ. Лаборатория знаний, 2007. — 224 с.

6. Персональный компьютер: настройка и техническая поддержка : методич. пособие для учителя и ЭОР на компакт-диске. — 3-е изд. — М. : БИНОМ. Лаборатория знаний, 2007. — 79 с.

7. Учебные проекты с использованием Microsoft Office : учеб. пособие. — 3-е изд., испр. — М. : БИНОМ. Лаборатория знаний, 2007. — 230 с.

8. Учебные проекты с использованием Microsoft Office : методич. пособие и ЭОР на компакт-диске. — 3-е изд. — М. : БИНОМ. Лаборатория знаний, 2007. — 93 с.

Библиотека программирования

1. *Аньен Ф.* ASP.NET с примерами на Visual Basic. NET / Ф. Аньен. — М. : БИНОМ. Лаборатория знаний, 2005. — 400 с.

2. *Бишоп Дж.* C# в кратком изложении / Дж. Бишоп. — М. : БИНОМ. Лаборатория знаний, 2005. — 472 с.

3. *Великович Л.С.* Программирование для начинающих / Л.С. Великович, М.С. Цветкова. — М. : БИНОМ. Лаборатория знаний, 2007. — 287 с.

4. *Желонкин А.В.* Основы программирования в интегрированной среде DELPHI : практикум / А.В. Желонкин. — 2-е изд. — М. : БИНОМ. Лаборатория знаний, 2006. — 236 с.

5. *Златопольский Д.М.* Программирование : типовые задачи, алгоритмы, методы / Д.М. Златопольский. — М. : БИНОМ. Лаборатория знаний, 2007. — 223 с.

6. *Кирюхин В.М.* Методика решения задач по информатике. Международные олимпиады / В.М. Кирюхин, С.М. Окулов. — М. : БИНОМ. Лаборатория знаний, 2007. — 600 с.

7. *Конгер Д.* Физика для разработчиков компьютерных игр / Д. Конгер ; пер. с англ. А. С. Молявко. — М. : БИНОМ. Лаборатория знаний, 2007. — 520 с.

8. *Лесневский А.С.* Объектно-ориентированное программирование для начинающих / А. Лесневский. — М. : БИНОМ. Лаборатория знаний, 2005. — 232 с.

9. *Окулов С.М.* Задачи по программированию / С. М. Окулов. — М. : БИНОМ. Лаборатория знаний, 2006. — 820 с.

10. *Окулов С.М.* Основы программирования / С. М. Окулов. — 4-е изд. — М. : БИНОМ. Лаборатория знаний, 2008. — 440 с.

11. *Окулов С.М.* Программирование в алгоритмах / С. М. Окулов. — 3-е изд. — М. : БИНОМ. Лаборатория знаний, 2007. — 383 с.

12. *Окулов С.М.* Ханойские башни : занятия по информатике с одаренными школьниками / С. М. Окулов, А. В. Лялин. — М. : БИНОМ. Лаборатория знаний, 2008. — 245 с.

13. *Плаксин М.А.* Тестирование и отладка программ для профессионалов будущих и настоящих / М. А. Плаксин. — М. : БИНОМ. Лаборатория знаний, 2007. — 167 с.

14. *Робертсон А.А.* Программирование — это просто : пошаговый подход / А. А. Робертсон. — М. : БИНОМ. Лаборатория знаний, 2008. — 384 с.

15. *Стивенс Э.* Самоучитель по C++ / Э. Стивенс. — М. : БИНОМ. Лаборатория знаний, 2005. — 872 с.

16. *Столяр С.Е.* Информатика : представление данных и алгоритмы / С. Е. Столяр. — М. : БИНОМ. Лаборатория знаний, 2007. — 382 с.

Методические пособия «Информатизация образования» и «ИКТ в работе учителя»

1. Видеоматериалы и сетевые видеосервисы в работе учителя : практич. пособие / под ред. Я. С. Быховского. — М. : БИНОМ. Лаборатория знаний, 2008. — 90 с.

2. *Драхлер А.Б.* Сеть творческих учителей / А. Б. Драхлер. — М. : БИНОМ. Лаборатория знаний, 2008. — 171 с.

3. Модели управления процессами комплексной информатизации общего среднего образования / М. С. Цветкова [и др.]. — М. : БИНОМ. Лаборатория знаний, 2005. — 111 с.

4. *Пожарина Г.Ю.* Стратегия внедрения свободного программного обеспечения в учреждениях образования / Г. Ю. Пожарина, А. М. Поносов. — М. : БИНОМ. Лаборатория знаний, 2008. — 152 с.

5. *Сайков Б.П.* Организация информационного пространства образовательного учреждения / Б. П. Сайков. — М. : БИНОМ. Лаборатория знаний, 2005. — 406 с.

6. *Спиридонов О.В.* Microsoft Word. От пользователя к специалисту : методич. пособие и ЭОР на компакт-диске / О. В. Спиридонов, Н. С. Вольнян. — М. : БИНОМ. Лаборатория знаний, 2008. — 356 с.

ОГЛАВЛЕНИЕ

Предисловие 3

Глава 1. **Информационная деятельность человека** 4
1.1. Основные этапы информационного развития общества 4
1.2. Роль информационной деятельности в современном обществе 7
1.3. Информационные ресурсы общества 21

Глава 2. **Информация и информационные процессы** 30
2.1. Информация и ее свойства 30
2.2. Информация и управление 34
2.3. Информация и моделирование 38
2.4. Структурные информационные модели 42
2.5. Пример построения математической модели 47
2.6. Единицы измерения информации 48
2.7. Системы счисления 50
2.8. Модель перевода чисел из одной системы счисления в другую 54
2.9. Двоичная и шестнадцатеричная системы счисления как модель представления чисел в компьютере 57
2.10. Двоично-кодированные системы 62
2.11. Кодирование информации 63
2.12. Файловая система хранения, поиска и обработки информации на диске 73
2.13. Основы алгоритмизации 89
2.14. Примеры алгоритмов обработки информации 102
2.15. Системы и технологии программирования 105
2.16. Введение в язык программирования. Синтаксис программы 110
2.17. Введение в язык программирования. Семантика программы 120

Глава 3. **Средства информационных и коммуникационных технологий** 128
3.1. История компьютера 128
3.2. Состав персонального компьютера 138
3.3. Логические функции и схемы — основа элементной базы компьютера 148
3.4. Логические выражения и таблицы истинности. Логические схемы и логические диаграммы 156

3.5. Программное обеспечение персонального компьютера 162
3.6. Защита информации 169
Глава 4. **Технологии создания и преобразования информационных объектов** 177
4.1. Технология обработки текстовой информации 177
4.2. Текстовый процессор 184
4.3. Использование шаблонов документов и других средств, повышающих эффективность работы с текстом 193
4.4. Программы для верстки оригинал-макетов 196
4.5. Технология обработки графической информации 199
4.6. Графика в профессии 203
4.7. Видеомонтаж 209
4.8. Автоматизированное проектирование 213
4.9. Технология обработки звуковой информации 217
4.10. Синтезаторы звука на компьютере 220
4.11. Система компьютерной презентации 226
Глава 5. **Технологии работы с информационными структурами — электронными таблицами и базами данных** 232
5.1. Компьютер как вычислитель 232
5.2. Моделирование электронной таблицы 237
5.3. Примеры моделирования в электронной таблице 254
5.4. База данных как модель информационной структуры 261
5.5. Компьютерная база данных — система организации, хранения, доступа, обработки и поиска информации 267
5.6. Модель расчета оплаты труда в табличной базе данных 280
Глава 6. **Телекоммуникационные технологии** 286
6.1. Компьютерная сеть как средство массовой коммуникации 286
6.2. Локальная вычислительная сеть 292
6.3. Интернет-страница и редакторы для ее создания 298
6.4. Личные сетевые сервисы в Интернете 307
6.5. Коллективные сетевые сервисы в Интернете 316
6.6. Пример работы в телеконференции на основе Skype 323
6.7. Сетевая этика и культура 330
Заключение **335**
Приложение. Список открытых интернет-ресурсов **339**
Список литературы **343**

Учебное издание

Цветкова Марина Серафимовна
Великович Леонид Соломонович

Информатика и ИКТ

Учебник

6-е издание, стереотипное

Редактор *Л. В. Толочкова*
Технический редактор *Н. И. Горбачёва*
Компьютерная верстка: *Г. Ю. Никитина*
Корректор *А. П. Сизова*

Изд. № 106114468. Подписано в печать 07.02.2014. Формат 60 × 90/16.
Гарнитура «Петербург». Бумага офс. № 1. Печать офсетная.
Усл. печ. л. 22.5 (в т.ч. цв. вкл. 0,5). Тираж 10 000 экз. Заказ № 35331.

ООО «Издательский центр «Академия». www.academia-moscow.ru
129085, Москва, пр-т Мира, 101В, стр. 1.
Тел./факс: (495) 648-0507, 616-00-29.
Санитарно-эпидемиологическое заключение № РОСС RU. АЕ51. Н 16476 от 05.04.2013.

Отпечатано в соответствии с качеством предоставленных издательством
электронных носителей в ОАО «Саратовский полиграфкомбинат».
410004, г. Саратов, ул. Чернышевского, 59. www.sarpk.ru