

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Комсомольский-на-Амуре государственный технический университет»  
Институт новых информационных технологий  
Федерального государственного бюджетного образовательного учреждения  
высшего профессионального образования  
«Комсомольский-на-Амуре государственный технический университет»

**А.А. Просолович**

## **КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ И ИНФОРМАТИКА**

Утверждено в качестве учебного пособия ученым советом  
Федерального государственного бюджетного образовательного учреждения  
высшего профессионального образования  
«Комсомольский-на-Амуре государственный технический университет»

Комсомольск-на-Амуре 2012

УДК 004.38:002 (07)  
ББК 32.973 я7  
П 826

**Просолович, А.А.**

П 826 Компьютерные технологии и информатика: учебное пособие / А.А. Просолович. – Комсомольск-на-Амуре: ФГБОУ ВПО «КНАГТУ», 2012. – 130 с.

Изложены основные понятия информатики и информационных технологий в контексте формирования информационного общества. Рассмотрены базовые компьютерные технологии, их возможности и особенности. Подробно рассмотрена технология баз данных на примере СУБД MS Access.

Предназначено для студентов направления 031600 «Реклама и связи с общественностью» заочной формы обучения с использованием дистанционных технологий.

ББК 32.973 я7

- © Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Комсомольский-на-Амуре государственный технический университет», 2012
- © Институт новых информационных технологий Федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Комсомольский-на-Амуре государственный технический университет», 2012

## ВВЕДЕНИЕ

Во второй половине XX в. человечество вступило в новый этап своего развития – начался переход от индустриального общества к информационному. Процесс, обеспечивающий этот переход, получил название информатизации. *Информатизация* – это процесс создания, развития и всеобщего применения информационных средств и технологий, обеспечивающих достижение и поддержание уровня информированности всех членов общества, необходимого и достаточного для кардинального улучшения качества труда и условий жизни в обществе. При этом информация становится важнейшим стратегическим ресурсом общества и занимает ключевое место в экономике, образовании и культуре.

Неизбежность информатизации обусловлена резким возрастанием роли и значения информации. Информационное общество характеризуется высокоразвитой информационной сферой, которая включает деятельность человека по созданию, переработке, хранению, передаче и накоплению информации.

Научным фундаментом процесса информатизации общества служит новая научная дисциплина – информатика.

# 1. ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

## 1.1. ЧЕЛОВЕК И ИНФОРМАЦИЯ

### 1.1.1 ТЕРМИНОЛОГИЯ ИНФОРМАТИКИ

Термин «*информация*» имеет множество определений. В законе об информации, информационных технологиях и о защите информации №149-ФЗ от 27 июля 2006 года (с изменениями от 27 июля 2010 года) дано следующее определение: информация – сведения (сообщения, данные) независимо от формы их представления.



Рис. 1.1. Общая схема передачи информации

В широком смысле информация – это отражение реального мира; в узком смысле – это любые сведения, являющиеся объектом хранения, передачи и преобразования.

Понятие «информация» вместе с понятиями «вещество» и «энергия» относится к основным понятиям науки. С практической точки зрения информация всегда представляется в виде сообщения. Информационное сообщение связано с источником сообщения, получателем сообщения и каналом связи (рис. 1.1).

Сообщение от источника к приемнику передается в материально-энергетической форме (электрической, световой, звуковой и т.д.). Человек воспринимает сообщения посредством органов чувств. Приемники информации в технике воспринимают сообщения с помощью различной измерительной и регистрирующей аппаратуры. В обоих случаях с приемом информации связано изменение во времени какой-либо величины, характеризующей состояние приемника.

В этом смысле информационное сообщение можно представить функцией  $x(t)$ , характеризующей изменение во времени материально-энергетических параметров физической среды, в которой осуществляются информационные процессы.

Функция  $x(t)$  принимает любые вещественные значения в диапазоне изменения времени  $t$ . Если функция  $x(t)$  непрерывна, то имеется непрерывная (или аналоговая) информация, источником которой обычно являются различные природные объекты (например, температура, давление, влажность воздуха), объекты технологических производственных процессов (например, нейтронный поток в активной зоне, давление и температура теплоносителя в контурах ядерного реактора). Если функция  $x(t)$  дискретна, то информационные сообщения, используемые человеком, имеют характер дискретных сообщений (например: сигналы тревоги, передаваемые посредством световых и звуковых сообщений; языковые сообщения, передаваемые в письменном виде или с помощью звуковых сигналов; сообщения, передаваемые с помощью жестов, и др.).

В современном мире информация, как правило, обрабатывается на вычислительных машинах. Поэтому информатика тесно связана с инструментарием – вычислительной машиной.

**Компьютер** (англ. *computer* – «вычислитель») – многозначный термин, наиболее часто употребляется в качестве обозначения программно управляемого электронного устройства обработки информации. Термин «компьютер» и аббревиатура «ЭВМ» (электронная вычислительная машина), принятая в русскоязычной научной литературе, являются синонимами.

Электронная вычислительная машина – вычислительная машина, построенная с использованием в качестве функциональных элементов электронных устройств вместо механических. Термин употреблялся для отличия от исторического предшественника – механической вычислительной машины.

В настоящее время словосочетание «электронная вычислительная машина» почти вытеснено из бытового употребления. Аббревиатуру «ЭВМ» в основном используют как правовой термин в юридических документах, инженеры цифровой электроники, также в историческом смысле – для обозначения компьютерной техники 1940-1980-х годов, и для обозначения больших вычислительных устройств, в отличие от персональных.

В зависимости от вида перерабатываемой информации вычислительные машины (ВМ) подразделяют на два основных класса: аналоговые и цифровые (рис. 1.2).

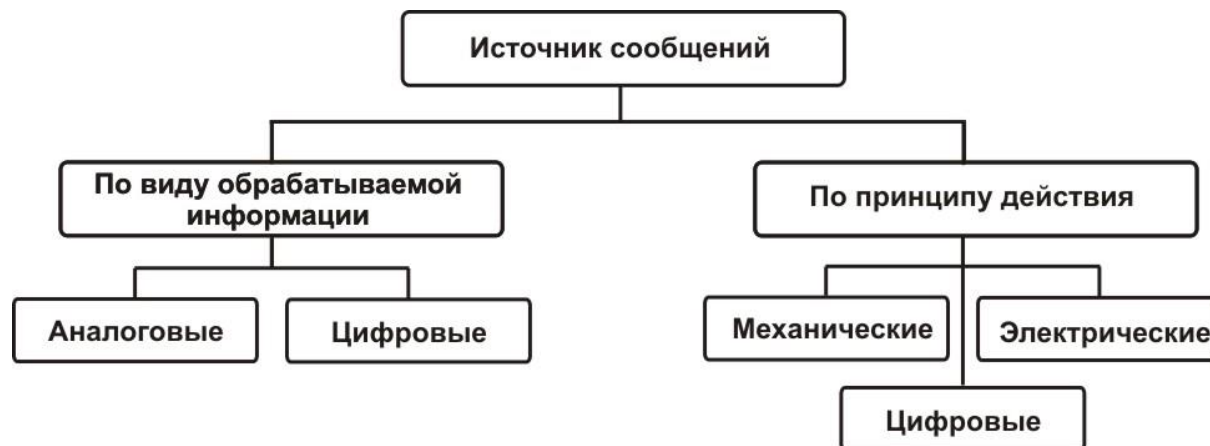


Рис. 1.2. Классы вычислительных машин

Аналоговая вычислительная машина (АВМ) – это машина, оперирующая информацией, представленной в виде непрерывных изменений некоторых физических величин. При этом в качестве физических переменных выступают сила тока электрической цепи, угол поворота вала, скорость и ускорение движения тела и т. п. Так как многие явления в природе математически описываются одними и теми же уравнениями, на АВМ с помощью одного физического процесса можно моделировать различные процессы.

Цифровая вычислительная машина (ЦВМ) – это машина, оперирующая информацией, представленной в цифровом виде. В настоящее время разработаны методы численного решения многих видов уравнений, что дало возможность решать на ЦВМ различные уравнения и задачи с помощью набора простых арифметических и логических операций. Поэтому если АВМ обычно предназначены для решения определенного класса задач, т.е. являются специализированными, то ЦВМ, как правило, – универсальное вычислительное средство. Наибольшее распространение получили электронные вычислительные машины, выполненные с использованием новейших достижений электроники.

Очень широко употребляется еще один термин – «**данные**» (от лат. *data*). Его принято применять в отношении информации, представленной в виде, позволяющем хранить, передавать или обрабатывать ее с помощью технических средств. Поэтому наряду с терминами «ввод информации», «обработка информации», «хранение информации», «поиск информации» используются термины «ввод данных», «обработка данных», «хранение данных» ит. п.

Информация существует не только в виде данных, но и в виде знаний. Здесь **знания** – это совокупность объективных фактов, способов и технологий, систематизированных и дающих реальное представление о предметах, процессах и явлениях, т.е. специальным образом структурированная информация.

Данные и знания в виде информационных потоков циркулируют в **информационных системах** (ИС). Сбор, накопление, обработка, хранение и использование информации в ИС осуществляются с помощью информационных технологий. **Информационные технологии** (ИТ) – это механизированные (инженерные) способы обработки информации, которые реализуются посредством **автоматизированных информационных систем** (АИС).

Объектом информатики выступают автоматизированные (человеко-машинные), основанные на компьютерах и телекоммуникационной технике информационные системы различных классов и назначения. Поэтому рассмотрим более подробно объект и предмет информатики.

### 1.1.2. ОБЪЕКТ И ПРЕДМЕТ ИНФОРМАТИКИ

Автоматизированные информационные системы как объект информатики получили широчайшее распространение. В качестве признаков классификации АИС выступают область применения, охватываемая территория, организация информационных процессов, направление деятельности, назначение, структура и др. Классификация АИС по направлению деятельности показана на рис 1.3.

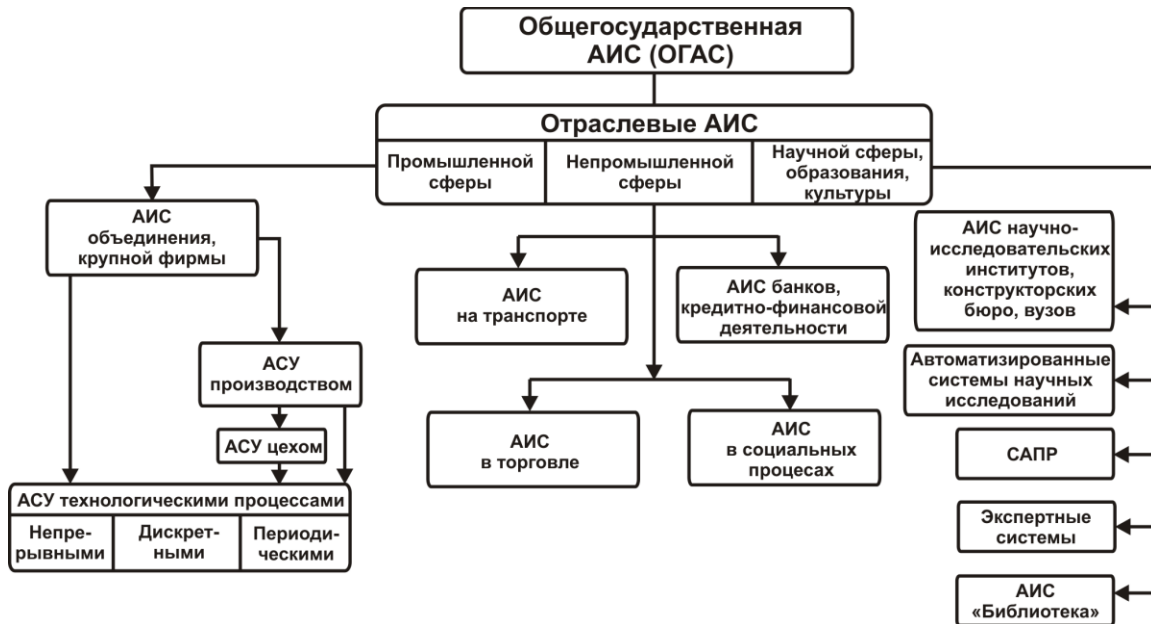


Рис. 1.3. Классификация АИС по направлению деятельности

В промышленной сфере преобладает отраслевой характер иерархии АИС. Классификация АИС по территориальному признаку приведена на рис. 1.4.

В зависимости от организации информационных процессов АИС подразделяют на управляющие и информационные. В информационных системах управление отсутствует (автоматизированные системы научных исследований – АСНИ, «Библиотека», системы автоматизированного проектирования – САПР, экспертные системы – ЭС и др.).

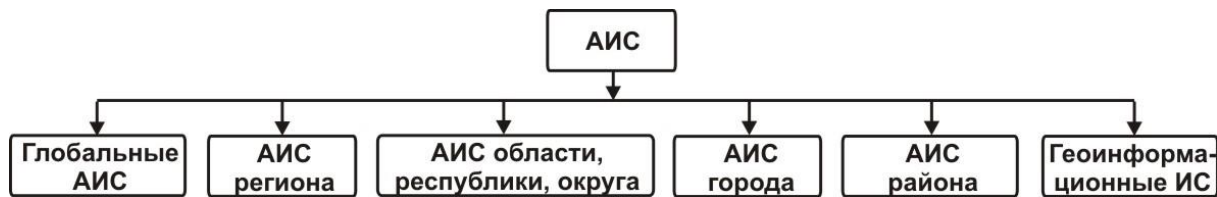


Рис. 1.4 – Классификация АИС по территории

По области применения АИС классифицируют на административные, производственные, учебные, медицинские, военные, метеорологические, экологические, криминалистические и др. В данном случае назначение и структура построения АИС характеризуются наличием соответствующих подсистем (рис. 1.5). Этот класс АИС исторически является одним из первых на производстве.

Автоматизированные информационные системы различают также по уровню развития – в зависимости от поколений компьютеров, на которых они базируются. Разно-

образе ИТ и АИС постоянно растет. Следует выделить ИТ и АИС, с помощью которых регулируются микросоциальные процессы. Это ИТ и АИС денежно-кассовых операций, распределения мест на транспорте и в гостиницах, обработки метеорологической информации и т. п. Они также относятся к объекту информатики.

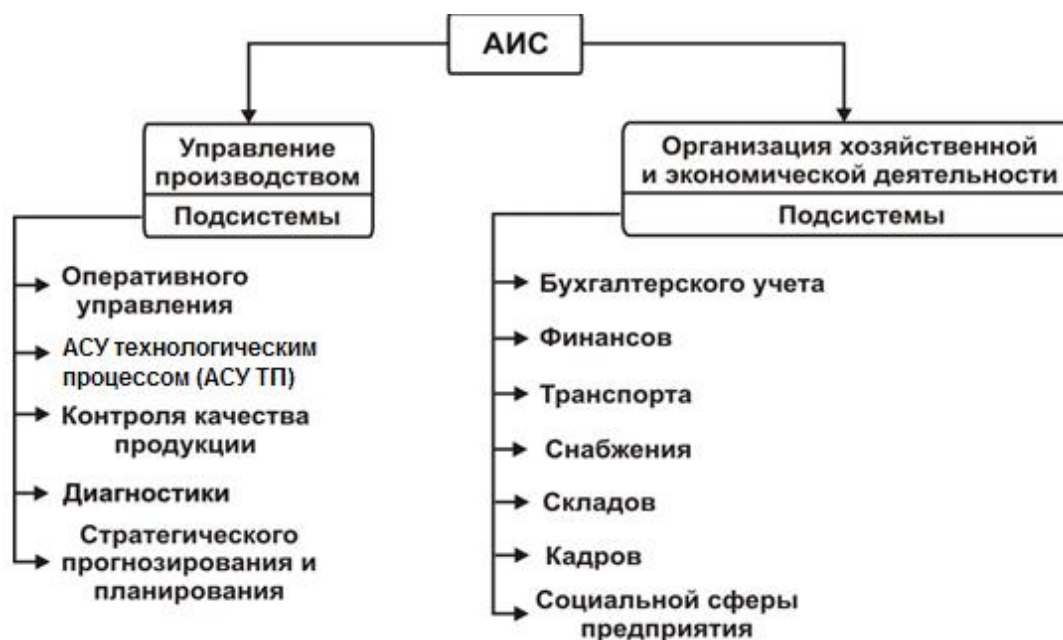


Рис. 1.5 – Классификация АИС на предприятии

Здесь необходимо особо отметить тот факт, что объектом информатики выступают не сами по себе компьютеры как программно-технические комплексы, а ИТ, воплощением которых служат компьютеризованные, или человеко-машинные, точнее социо-технические, системы. В отличие от производственных, энергопреобразующих технологий, ИТ (объект информатики) относится к социальным, знаниепреобразующим технологиям.

Переход к ИТ как технической базе автоматизированных информационно-управляющих систем обнажил и до крайности обострил проблемы, относящиеся ко всему технологическому циклу сбора, переработки и применения информации в плано-управленческих, познавательных и других процессах, выдвинул на первый план общие проблемы содержания информационных процессов и значения ИТ. Говорить о воздействии науки на что-либо вне информационного процесса бессмысленно. Научная идея должна превратиться в информацию, т.е. быть закодированной, переданной по каналу связи, принятой адресатом, чтобы ее можно было применить на практике. Чем больше потенциал знаний, тем важнее задача развития информационно-коммуникативных сфер экономики. Знания должны определенным образом фиксироваться, трансформироваться, распределяться, приниматься и перерабатываться.

Информационные технологии и выступили новым средством превращения знаний в **информационный ресурс** (ИР) общества, его новым движущим фактором, средством эффективного использования ИР. Информационный ресурс наряду с другими ресурсами общества (материальными, финансовыми, природными, трудовыми и др.) стал основным ресурсом человечества, главной ценностью современной цивилизации. Но возникли и сложные проблемы, относящиеся к роли, механизму функционирования, социальным последствиям использования ИР. Для их решения и появилась новая наука – информатика.

**Предметом информатики** как новой фундаментальной науки выступает информационный ресурс – его сущность, законы функционирования, механизмы взаимодействия с другими ресурсами общества и воздействия на социальный прогресс. Переход на уровень ИР в его содержательной трактовке означает переход к изучению внутренних связей и закономерностей, основанных на использовании ИТ.

### 1.1.3. ВИДЫ, ОСОБЕННОСТИ И ФОРМЫ ИНФОРМАЦИОННОГО РЕСУРСА

Понятие информационного ресурса сводится к следующему: ИР – это знание, обладающее всеми атрибутами понятия «информация»; и наоборот: ИР – это информация, которая имеет все свойства понятия «знание». Можно сказать и так: ИР – это информация, т.е. информация в виде понятийного знания.

Принципиальное значение в понимании ИР имеет форма существования и представления знаний. **Знание** – это отражение той или иной стороны объективной действительности в виде идей, понятий, представлений о каком-то предмете или явлении. Здесь речь идет о понятийном знании – знании в смысловом выражении. Имеется, как известно, и другой вид знаний – знания интуитивные, рожденные в подсознательной сфере человека, которые, прежде чем стать сообщениями, должны быть выражены в виде понятий.

Знания, будучи живой, диалектической системой, передаются другим людям, материализуются и существуют в трех видах:

- 1) «живые» знания (квалификация);
- 2) овеществленные знания;
- 3) информация (сообщения).

Особо следует отметить, что не все знания, в частности книги, газеты, радио- и телепередачи, патенты и т. п., выступают в качестве ИР. Появляется новая проблема – информативность сообщений. Книги, патентные описания и другие сообщения еще должны найти своих потребителей. Из них еще должна быть извлечена информация. Перед человечеством стоит громадная по своей важности и сложности задача: извлечь максимум информации из сообщений, накопленных за всю историю человечества, и превратить ее в активно функционирующий ресурс. Речь идет о превращении книжных описаний и других «рассеянных» знаний в алгоритмы и программы – это часть работ по формированию ИР.

Главная трудность в понимании природы и функций ИР как интеллектуального ресурса, фактора коллективного творчества состоит в раскрытии механизма перехода знаний в силу, способов его воздействия на материальные факторы прогресса.

Но не сам по себе ИР «движет» системы. Движителями систем выступают материальные силы: энергетические и трудовые факторы. Информационный ресурс же, словно фермент, переводит эти материальные факторы из латентного в активное состояние и вводит их в заданное русло, обеспечивая, тем самым, прирост свободной энергии в целенаправленных системах социальной природы за счет снижения их энтропии.

Каковы же основные особенности информационного ресурса?

**Первая особенность ИР:** в отличие от других видов ресурсов (в частности, материальных), он практически неисчерпаем; по мере развития общества и роста потребления знаний запасы ИР не убывают, а растут.

**Вторая особенность ИР:** по мере использования он не исчезает, а сохраняется и даже увеличивается (за счет трансформации полученных сообщений с учетом опыта и местных условий).

**Третья особенность ИР:** он несамостоятелен и сам по себе имеет лишь потенциальное значение. Только соединяясь с другими ресурсами (опытом и квалификацией персонала, техникой, энергией, сырьем), ИР проявляется как движущая сила.

**Четвертая особенность ИР:** эффективность его применения связана с эффектом повторного производства знаний. По словам К. Маркса, продукт умственного труда – наука – всегда ценится далеко ниже ее стоимости, потому что рабочее время, необходимое для ее восприятия, не идет ни в какое сравнение с тем рабочим временем, которое требуется для того, чтобы первоначально ее произвести. Так, например, теорему о биноме школьник может выучить в течение одного часа. Информационное взаимодействие позволяет получить новое знание ценой значительно меньших затрат по сравнению с затратами труда, энергии, времени, затраченными на его прямое генерирование.

**Пятая особенность ИР:** он является формой непосредственного включения науки в состав производительных сил. В индустриальном обществе наука выступает опосредованной производительной силой. Посредниками при этом служат машины,

новые материалы и препараты. В информационном обществе наука представляет собой непосредственную производительную силу, выступая в форме ИР. Радикально изменяется характер действия ИР: производительность труда повышается не на проценты, а на порядки – в 10–100 раз. Осуществляется переход к производственным системам, основной функционирования которых служат ИР и информационные связи.

**Шестая особенность ИР:** он возникает в результате не просто умственного труда, а его творческой части. Рутинная часть умственной работы сама по себе не информативна: она не увеличивает потенциала нужных знаний, не меняет представления о путях достижения цели. Любой умственный труд, будь то наука или управление, включает две части: рутинную и творческую. Увеличение умственной работы за счет рутинной ее части не ведет к росту ИР. Вместе с тем трудовые и материально-энергетические затраты необходимы при выработке, передаче и использовании ИР.

Существует две формы ИР как отчуждаемых знаний, становящихся сообщениями: пассивная и активная. К **пассивным** формам ИР относятся книги, журнальные статьи, патенты и базы данных, а также знания, привязанные к конкретным предметным областям (например, выборки, извлечения данных и т. п.), если они не комплексные, т.е. недостаточны для целенаправленного применения. **Активные** формы ИР: модель, алгоритм, программа, проект и база знаний. Их можно трактовать в целом как стадии созревания ИР, степени доведения его до готовности и превращения в силу. Естественно, что эти формы ИР различаются научно-техническим уровнем и завершенностью.

**Модель** – это описание системы, отображающее определенную группу ее свойств. Создание модели системы позволяет предсказывать ее поведение в определенном диапазоне условий.

**Алгоритмы** подразделяют в зависимости от степени общности. Важно стремиться к созданию решающих алгоритмов. **Программа и проект** – конечные, синтетические формы существования ИР в его жизненном цикле. В виде программы и проекта, а часто в виде решающего алгоритма ИР непосредственно противостоит энтропии рассматриваемого объекта. В этом плане вводится понятие информационной емкости программы, которое означает величину потенциального уменьшения неопределенности. В ходе реализации программы идет как бы заполнение пространства объекта информацией, которая в нем сконцентрирована.

Модель, алгоритм, программа, проект и особенно база знаний (БЗ) как активные формы ИР – это антиэнтропийные инструменты. Однако программа и проект выделяются среди них своей закономерностью и готовностью к прямому информационному воздействию на объект с целью снятия его неопределенности.

Теперь можно дать определение понятия «информатика». **Информатика** – это научная дисциплина, изучающая закономерности, методы и способы накопления, хранения, передачи и использования информации с помощью компьютеров и других технических средств.

Информатика как наука о законах получения, передачи и использования ИР в общественной практике подводит теоретический фундамент под использование компьютеров в автоматизированных системах, которые и предназначены для усиления информационных процессов в обществе, для использования ИР. Речь идет прежде всего о специальных ИР, основанных на компьютерной технике и реализующих ИТ, т.е. инженерную обработку знаний (Knowledge Engineering).

#### 1.1.4. КРАТКАЯ ИСТОРИЯ РАЗВИТИЯ ИНФОРМАТИКИ

Информатика как наука стала развиваться с середины XX столетия, что связано с появлением компьютера и начавшейся компьютерной революцией. Появление вычислительных машин в 1950-е гг. создало для информатики необходимую аппаратную поддержку, т.е. благоприятную среду для ее развития как науки. Всю историю информатики принято подразделять на два больших этапа: предысторию и историю.

Предыстория информатики такая же древняя, как и история развития человеческого общества. В предыстории также выделяют (весьма приближенно) ряд этапов. Каждый из них характеризуется резким возрастанием, по сравнению с предыдущим этапом, возможностей хранения, передачи и обработки информации.

**Начальный этап** предыстории информатики – освоение человеком развитой устной речи. Членораздельная речь, язык стали специфическим социальным средством хранения и передачи информации.

**Второй этап** – возникновение письменности. На этом этапе резко возросли возможности хранения информации. Человек получил искусственную внешнюю память. Организация почтовых служб позволила использовать письменность и как средство для передачи информации. Кроме того, возникновение письменности было необходимым условием для начала развития наук (вспомним, например, Древнюю Грецию). С этим же этапом, по всей видимости, связано и возникновение понятия «натуральное число». Все народы, обладавшие письменностью, владели понятием числа и пользовались той или иной системой счисления.

**Третий этап** – книгопечатание. Его можно смело назвать первой информационной технологией. Воспроизведение информации было поставлено на поток, на промышленную основу. На этом этапе, по сравнению с предыдущим, не столько увеличились возможности хранения информации (хотя и здесь был выигрыш: письменный источник – это часто один-единственный экземпляр, печатная книга – это целый тираж экземпляров, а, следовательно, и малая вероятность потери информации при хранении, как это случилось, например, с рукописью «Слова о полку Игореве»), сколько повысились доступность информации и точность ее воспроизведения.

**Четвертый (последний) этап** предыстории информатики связан с успехами точных наук (прежде всего математики и физики) и начинающейся научно-технической революцией. Этот этап характеризуется возникновением таких мощных средств связи, как радио, телефон и телеграф, а позднее и телевидение. Появились новые возможности получения и хранения информации – фотография и кино. К ним очень важно добавить разработку методов записи информации на магнитные носители (магнитные ленты, диски).

С разработкой первых компьютеров принято связывать возникновение информатики как науки, начало ее истории. Для такой привязки имеются следующие причины:

- во-первых, сам термин «информатика» появился благодаря развитию вычислительной техники, и поначалу под ним понималась наука о вычислениях (первые компьютеры большей частью использовались для проведения числовых расчетов);
- во-вторых, выделению информатики в отдельную науку способствовало такое важное свойство современной вычислительной техники, как **единая форма представления обрабатываемой и хранимой информации**. Вся информация независимо от ее вида хранится и обрабатывается на компьютере в двоичной форме. Так получилось, что компьютер в одной системе объединил хранение и обработку числовой, текстовой (символьной) и аудиовизуальной (звука, изображения, видео) информации. В этом состояла иницилирующая роль вычислительной техники при возникновении и оформлении новой науки.

На сегодняшний день информатика представляет собой комплексную научно-техническую дисциплину.

## 1.2. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

### 1.2.1. ЭТАПЫ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Если информация – категория всеобщая, присущая всем историческим периодам, то информатика – категория конкретно-историческая, присущая лишь современному и последующим историческим периодам. Ранее, когда не было ИТ, т.е. не было объекта информатики, информация перерабатывалась немашинными, рутинными способами.

На базе компьютеров рождаются ИС, реализующие специальные технологии сбора, переработки, передачи и применения информации.

Понятие технологии вообще включает комплекс научных и инженерных знаний, воплощенных в приемах труда, наборах материальных, технических, энергетических, трудовых факторов производства, способов их соединения для создания продукта или услуги, отвечающих определенным требованиям, стандартам. В таком понимании термин «технология» неразрывно связан с механизацией производственного или непромышленного (социального) процесса. Последнее важно подчеркнуть для обозначения точных исторических границ информатики: ее не могло быть в домашний период обработки и представления знаний, когда не было ИТ. Переработка информации с помощью компьютера, выработка новых знаний, соотношенных с целями пользователей, – функциональное назначение ИТ.

Информационная технология содержит:

- определенные наборы материальных средств (носители информации, технические средства измерения ее состояний, обработки, передачи и т. п.);
- способы их взаимодействия;
- совокупность определенных методов организации работы специалистов.

Информационная технология решения большого числа разных задач включает следующие важнейшие процедуры (которые могут быть сгруппированы по функционально-временным стадиям):

- сбор и регистрацию информации;
- передачу ее к месту обработки;
- машинное кодирование данных;
- хранение;
- поиск;
- вычислительную обработку;
- тиражирование информации;
- использование информации, т.е. принятие решений.

Как правило, информация подвергается всем процедурам преобразования, но в ряде случаев некоторые процедуры могут отсутствовать. Последовательность выполнения процедур также бывает различной, а некоторые процедуры могут повторяться. Состав процедур преобразования и особенности их выполнения во многом зависят от объема и системы автоматизированной обработки информации.

Рассмотрим особенности выполнения основных этапов преобразования информации.

**Процедура сбора и регистрация информации** осуществляется по-разному на различных объектах. Наиболее сложен этот этап в автоматизированных управленческих процессах таких промышленных предприятий, фирм и т. п., на которых происходят сбор и регистрация первичной учетной информации, отражающей производственно-хозяйственную деятельность объекта. Особое значение при этом придается достоверности, полноте и своевременности первичной информации.

На предприятии сбор и регистрация информации осуществляются при выполнении различных хозяйственных операций (при приеме готовой продукции, получении и отпуске материалов и т. п.). Сначала информацию собирают, затем ее фиксируют.

Учетные данные могут возникать на рабочих местах (например, в результате подсчета количества обработанных деталей, прошедших сборку узлов, изделий и т. п.). Для сбора фактической информации производятся измерения, подсчет, взвешивание материальных объектов, получение временных и количественных характеристик работы отдельных исполнителей.

Сбор информации, как правило, сопровождается регистрацией, т.е. фиксацией информации в бумажном документе или на машинном носителе. Запись в первичные документы в основном осуществляется вручную, поэтому процедура сбора и регистрации остается пока наиболее трудоемкой. В условиях автоматизации управления предприятием особое внимание придается использованию технических средств сбора и ре-

гистрации информации, ее накопления и передачи по каналам связи в компьютер с целью формирования первичного документа.

Необходимость передачи информации для различных объектов обосновывается по-разному. Так, в автоматизированной системе управления (АСУ) предприятием она вызвана тем, что процедура сбора и регистрации информации нередко территориально отделена от процедуры ее обработки. Сбор информации, как правило, осуществляется на рабочих местах, а регистрация – в вычислительном центре или на персональном компьютере.

**Передача информации** может производиться различными способами: с помощью курьера, пересылкой по почте, доставкой транспортными средствами, дистанционно по каналам связи. Последний способ сокращает время передачи данных. Для дистанционной передачи информации по каналам связи необходимы специальные технические средства. Некоторые технические средства сбора и регистрации автоматически собирают информацию с датчиков, установленных на рабочих местах, и передают ее в компьютер.

Дистанционно может передаваться как первичная информация с мест ее возникновения, так и результатная – в обратном направлении. В этом случае результатная информация отражается на различных устройствах: дисплеях, табло, печатных устройствах. Поступление информации по каналам связи в центр обработки в основном осуществляется двумя способами: на машинном носителе или непосредственно в компьютер с помощью специальных программных и аппаратных средств.

Дистанционная передача информации постоянно развивается и совершенствуется. Применение этого способа в многоуровневых системах значительно ускоряет прохождение информации с одного уровня управления на другой и сокращает общее время обработки данных.

**Машинное кодирование информации** – это процедура машинного представления (записи) информации на машинных носителях в кодах, принятых в компьютере. Такое кодирование производится путем переноса данных первичных документов на магнитные диски, информация с которых затем вводится в компьютер для обработки. В процессе записи информации на машинные носители возникает наибольшее количество ошибок, что объясняется трудоемкостью данной операции. Поэтому обязательно выполняются операции контроля.

**Обработка информации** на компьютере производится, как правило, централизованно, а на мини- и микроЭВМ – в местах возникновения первичной информации, где организуются автоматизированные рабочие места специалистов той или иной управленческой службы (отдела материально-технического снабжения и сбыта, отдела главного технолога, конструкторского отдела, бухгалтерии, планового отдела и т. п.).

Автоматизированное рабочее место (АРМ) специалиста включает персональный компьютер, работающий автономно или в вычислительной сети, и набор программных средств и информационных массивов для решения функциональных задач.

**Хранение и накопление информации** вызвано необходимостью ее многократного использования, применения постоянной информации и комплектации первичных данных до их обработки. Хранение осуществляется на машинных носителях в виде информационных массивов, где данные располагаются по установленному в процессе проектирования группировочному признаку.

**Поиск данных** – это выборка нужных данных из хранимой информации, включая поиск информации, подлежащей корректировке или замене. Процедура поиска выполняется на основе составленного запроса на нужную информацию.

Печать сводок может сопровождаться процедурой **тиражирования**, если документ с результатной информацией необходимо представить нескольким пользователям.

**Принятие решения** в автоматизированной системе организационного управления, как правило, осуществляется специалистом без применения технических средств, но на основе тщательного анализа результатной информации, полученной на компьютере. Процедура принятия решения осложняется тем, что специалисту приходится искать из

множества допустимых решений наиболее приемлемое, сводящее к минимуму потери ресурсов (временных, трудовых, материальных и т.д.). Благодаря применению компьютера повышается степень аналитичности обрабатываемых сведений, а также обеспечивается постепенный переход к автоматизации выработки оптимальных решений в процессе диалога пользователя с вычислительной системой. Математическая теория принятия решений (ТПР) строится на основе теории игр и исследования операций.

### 1.2.2. ИНФОРМАЦИОННЫЕ ОСНОВЫ ПРОЦЕССОВ УПРАВЛЕНИЯ

Понятие «управление» – одно из центральных понятий кибернетики, науки о наиболее общих закономерностях в технических, биологических и социальных системах. Управление пронизывает все сферы деятельности человека и общества.

**Управление** – это целенаправленная организация того или иного процесса, протекающего в системе. В общем случае процесс управления состоит из следующих четырех элементов:

- 1) получения информации о задачах управления (цели управления);
- 2) получения информации о результатах управления (т.е. о поведении объекта управления);
- 3) анализа полученной информации и выработки решения;
- 4) использования решения (т. е. осуществления управляющих воздействий).

Процесс управления – это информационный процесс, который заключается в следующем:

- сборе информации о ходе процесса;
- передаче ее в пункты накопления и переработки;
- анализе поступающей, накопленной и справочной информации;
- принятии решения на основе выполненного анализа;
- выработке соответствующего управляющего воздействия и доведении его до объекта управления.

Каждая фаза процесса управления протекает во взаимодействии с окружающей средой при воздействии различного рода помех. Цели, принципы и границы управления зависят от сущности решаемой задачи.

Управление всегда «разыгрывается» между двумя объектами – объектом управления и управляющим органом. Объект управления служит для удовлетворения материальных потребностей человека, т.е. для производства материальных благ (предметов потребления – продуктов промышленности и сельского хозяйства, транспортных средств, электричества, тепла, средств коммуникации и т.д.), а также для удовлетворения интеллектуальных потребностей (кино, телевидение, радио, образование, театр и т.д.).

Через объект управления проходят в основном материальные потоки и в значительно меньшей степени – информационные. В то же самое время через управляющий орган проходят только информационные потоки. Именно поэтому процесс управления является информационным процессом, а все люди, занятые в сфере управления, имеют дело только с информацией. Отсюда вторым важным понятием является понятие «система управления».

**Система управления (СУ)** – это совокупность взаимодействующих между собой объекта управления и органа управления, деятельность которых направлена на достижение заданной цели управления. В СУ решаются четыре основных типа задач управления:

- 1) **задача стабилизации системы** – поддержание ее выходных величин вблизи некоторых неизменных значений, несмотря на действие помех (стабилизацию напряжения и частоты тока в сети независимо от изменения потребления энергии);
- 2) **задача выполнения программы** – в случаях, когда заданные значения управляемых величин изменяются во времени заранее известным образом (полет ракеты, выполнение работ по заранее полученному графику);

3) *задача слежения* – это как можно более точное соответствие между текущим состоянием и состоянием другой системы (управление производством в условиях изменения спроса; слежение за целью – самолетом, кораблем);

4) *задача оптимизации* – наилучшим образом выполнение поставленной перед системой задачи при заданных реальных условиях и ограничениях.

### 1.2.3. ИНФОРМАТИЗАЦИЯ ОБЩЕСТВА

В истории человечества было несколько промышленных и информационных переворотов. Первый промышленный переворот основывался на паровом двигателе, затем на электромоторе и двигателе внутреннего сгорания, т.е. на энергопреобразующих машинах. С ним связаны индустриализация общества, переход к индустриальной цивилизации. Вторым промышленным переворотом связан с овладением человечеством атомной энергией и проникновением в космос – современная научно-техническая революция (НТР).

Перевороты в информационной сфере были уже трижды. При первом перевороте – появлении письменности – возник специальный механизм фиксирования и распространения информации во времени и пространстве. Появились документирование информации и информационно-накопительные центры в виде библиотек, архивов. Вторым информационным переворотом связан с книгопечатанием – тиражированием документированной информации с помощью специальной машины – печатного станка. Если появление письменности коррелирует с ростом науки, просвещения и искусства, то типографский станок породил книжно-журнальную и газетную индустрию – основу дальнейшего культурного прогресса человечества, а также индустрию документооборота. Прорыв состоял в производстве «бумажной» информации, возникновении механизма ее размножения и хранения. Но переработка и использование информации по-прежнему зависели от физиологических возможностей человека – от его способности за секунду воспринимать восемь-девять единиц информации и в лучшем случае произвести 15 логических операций.

Скачкообразное увеличение скорости распространения информации, вызванное появлением электрического телеграфа, телефона, радио, телевидения, а также ростом на два-три порядка транспортных скоростей в связи с появлением автотранспорта и авиации, конечно, привело к огромной интенсификации информационных потоков и коммуникативных процессов (третий информационный переворот). Но это существенно не отразилось на скорости выработки сигналов обратной связи, которые по-прежнему зависели от способностей «невооруженной» головы человека.

Таким образом, человечество столкнулось с новым противоречием – между машинной вооруженностью рук (в середине XX в. 99 % всей физической работы выполнялось машинами) и «ремесленной вооруженностью» головы (интеллекта). Информационно-управляющие возможности людей, ограниченные физическими рамками, перестали соответствовать вещественно-энергетическим возможностям производства, ставшим по существу безграничными в силу использования машинной техники. Постепенно разрастаясь, это противоречие вело к информационно-организационному кризису, который и разрешился путем создания информационных, компьютеризованных технологий в области переработки и использования знаний.

С первым промышленным переворотом связано появление промышленных технологий, со вторым – появление информационных технологий, насыщение которыми народного хозяйства и означает информатизацию общества. Переворот в информационной сфере, ее индустриализация на базе компьютеров составляют главное содержание современной НТР; в то же время его можно считать третьим (после изобретения письменности и книгопечатания) в истории великим информационным переворотом.

До начала 1970-х гг., когда компьютеризация была поверхностной и очаговой, потребность в информатизации общества остро не ощущалась. Резкое расширение процессов компьютеризации, массовое развитие микропроцессорной техники и особенно

наметившийся переход к пятому поколению компьютеров сделали информатизацию насущно необходимой. Информатизация – это процесс, происходящий на базе компьютеризации, т.е. индустриализации непромышленной сферы народного хозяйства, процесс формирования новой, автоматизированной среды зарождения знаний, их переработки, распространения и превращения в силу, в материальный фактор.

Техническая база информатизации – это компьютерные и телекоммуникационные системы и сети, которые составляют «ядро» экономики, точнее – производственного аппарата будущего общества. Такой аппарат будет включать роботы и роботизированные производства, обрабатывающие центры, гибкие производственные системы, безлюдные участки, цехи, предприятия и, конечно, новые организационно-управленческие комплексы и системы связи.

Альтернативы информатизации нет. Это объективный этап социального прогресса во всех областях, прежде всего в экономике, управлении, науке и технологии.

В настоящее время стало ясно: для того, чтобы та или иная страна могла занять достойное место в мире и на равных участвовать в экономическом соревновании с другими государствами, она должна перестраивать и приспособлять свои структуры, приоритеты, ценности, институты к требованиям индустриальной ИТ. Экономические позиции той или иной страны вплоть до середины XXI века будут определяться такими новаторскими технологиями, как термоядерный синтез, биотехнология, плазменные процессы, космическая связь и др. Но их развитие, в свою очередь, зависит от развития информатики. Нельзя представить себе эффективное и надежное функционирование, например, атомных станций или космической связи без высокотехнологичного, т.е. компьютерного, информационно-организованного, обеспечения.

Информационное общество характеризуется следующими **основными признаками**:

- 1) большинство работающих в информационном обществе (около 80 %) заняты в информационной сфере, т.е. сфере производства информации и информационных услуг;
- 2) обеспечены техническая, технологическая и правовая возможности доступа любому члену общества практически в любой точке территории и в приемлемое время к нужной ему информации (за исключением военных и государственных секретов, оговоренных в соответствующих законодательных актах);
- 3) информация становится важнейшим стратегическим ресурсом общества и занимает ключевое место в экономике, образовании и культуре.

Информатизация – необходимое условие научно-технического, социального, экономического и политического прогресса в обществе. Неизбежность информатизации обусловлена:

- беспрецедентным усложнением социально-экономических процессов в результате увеличения масштабов и темпов общественного производства, углубления разделения труда и его специализации в условиях научно-технической революции;
- необходимостью адекватно реагировать на возникающие проблемы в динамично меняющейся обстановке, присущей постоянно развивающемуся обществу;
- повышением степени самоуправления предприятий, территорий, регионов.

Процесс перехода от постиндустриального общества к информационному происходит не одновременно в различных странах, он характеризуется также и разными темпами развития. Первыми на этот путь встали в конце 1950-х – начале 1960-х гг. США, Япония и страны Западной Европы. В этих государствах, начиная с 1960–1970-х гг., проводится политика повсеместной информатизации всех сфер деятельности человека. Были разработаны и приняты на государственном уровне программы информатизации с целью наиболее полного использования информационного ресурса для ускорения экономического, социального и культурного развития общества. Предполагается, что США завершат переход к информационному обществу к 2020 г., Япония и основные страны Западной Европы – к 2030–2040 гг.

В СССР в 1989 г. была разработана *Концепция информатизации общества*. По предварительным оценкам, информатизация в России завершится к 2050 г. при условии стабилизации экономической и политической обстановки в стране. По мнению специалистов, любая страна, насколько бы индустриально развитой она ни была, перейдет в разряд стран «третьего» мира, если опоздает с информатизацией.

В настоящее время объем расходов США на информатизацию (создание, производство, монтаж, использование компьютеров, информационных сетей и систем разного уровня, баз данных и т.д.) превышает объем военных расходов. Американские системы компьютерной оборонной инициативы (КОИ) и особенно стратегической оборонной инициативы (СОИ) базируются на развитых ИТ, которые, в свою очередь, предполагают информатизацию управления, науки, проектных разработок, всех систем выработки, принятия и поддержки решений.

Если предшествующие этапы развития человечества длились каждый около трех веков, то, по прогнозам ученых, информационный этап продлится значительно меньше. Срок его существования ограничится, вероятно, сотней лет. Это означает, что основные регионы мира войдут в развитое информационное общество в XXI в. и тогда же начнется переход к постинформационному обществу.

Для информационного общества характерно обеспечение требуемой степени информированности всех его членов, возрастание объема и уровня информационных услуг, предоставляемых пользователю. Информационное общество в теоретическом аспекте характеризуется высокоразвитой информационной сферой (инфосферой), которая включает деятельность человека по созданию, переработке, хранению, передаче и накоплению информации.

К первоочередным проблемам информатизации следует отнести психологическую проблему готовности населения к переходу в информационное общество. Этот переход в настоящее время затрудняется низким уровнем информационной культуры населения, недостаточной компьютерной грамотностью, а отсюда и низкими информационными потребностями и отсутствием желания их развивать. Наблюдается невосприимчивость экономики, управления на всех уровнях к результатам НТР, и в первую очередь в инфосфере. Психофизиологический аспект проблемы определяется совместимостью человека и новой информационной техники и технологии.

Информатизация, прежде всего, охватывает все элементы рыночной инфраструктуры экономики:

- сеть оптовой и розничной торговли;
- товарные, фондовые, валютные биржи, биржи труда, ярмарки;
- сети банков и кредитно-финансовых учреждений;
- сеть независимых посреднических фирм и контор, оказывающих различные услуги, в том числе и информационные, субъектам рынка;
- складское и транспортное хозяйство;
- страховые компании;
- налоговую службу;
- систему подготовки и переквалификации кадров;
- аудиторскую службу и т.д.

Для изучения спроса и предложения (товаров, идей, видов обслуживания) создаются региональные сети самых различных форм (лаборатории изучения спроса, информационно-коммерческие и рекламно-маркетинговые центры, центры аналитических исследований и научного прогнозирования и т.п.), специальные подразделения при крупных производственных, сбытовых и торговых предприятиях.

Важную роль в экономике играет сфера услуг. Она доминирует как в производстве валового продукта, так и в обеспечении занятости населения в регионе. Информатизация сферы услуг стимулирует развитие всех видов сервиса, особенно в торговле,

финансово-кредитной сфере, здравоохранении, организации досуга и отдыха населения, что положительно сказывается на производственной сфере, обеспечивает экономический рост, снижает напряженность на региональном рынке трудовых ресурсов. При этом возрастают потребности самой сферы услуг в информационных средствах, технологиях и услугах. Информатизация сферы услуг в экономике способна повысить производительность труда и тем самым обеспечить реальное безынфляционное увеличение доходов и рост благосостояния населения в течение длительного времени.

Информатизация бирж труда позволяет рационально использовать людские ресурсы региона, сбалансировать предложение и спрос на рабочую силу, предоставлять жителям региона работу в соответствии с их способностями и возможностями.

Информатизация проводится по различным направлениям экономики и социальной политики общества.

Таким образом, информационное общество – это общество, структуры, техническая база и человеческий потенциал которого приспособлены для оптимального превращения знаний в ИР и переработки последнего с целью перевода его пассивных форм (книг, статей, патентов и т.п.) в активные (модели, алгоритмы, программы, проекты). Но особое значение для активизации информационного потенциала общества имеет создание современных баз знаний. Это достигается на путях качественного преобразования традиционных баз данных (БД), рожденных до появления искусственного интеллекта, в базы знаний (БЗ).

Научным фундаментом процесса информатизации общества является новая научная дисциплина – информатика. В широком смысле информатика – это наука об информационной деятельности, информационных процессах и их организации в человеко-машинных системах. К основным разделам информатики относятся исследование и разработка информационных средств и технологий, программных средств, а также моделирование предметных областей.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Дайте определение понятия «информация».
2. В чем отличие знаний от данных?
3. Опишите общую схему системы передачи информации.
4. В чем разница между АВМ и ЦВМ?
5. Что такое АИС?
6. Раскройте понятие «информационная технология».
7. Дайте определение понятия «информационный ресурс».
8. Какие вы знаете виды ИР?
9. Каковы основные особенности ИР?
10. Перечислите и раскройте формы существования ИР.
11. Как вы понимаете «объект» и «предмет» информатики?
12. Сформулируйте определение информатики как науки.
13. Назовите этапы технологического процесса в информационных системах.
14. Дайте определение понятия «управление».
15. Почему процесс управления является информационным процессом?
16. Какие вы знаете информационные перевороты?
17. Каковы сущность и цели информатизации?
18. В чем состоит техническая база информатизации?
19. Раскройте основные признаки информационного общества.
20. Что является научным фундаментом процесса информатизации общества?

## 2. СИСТЕМЫ СЧИСЛЕНИЯ И ОСНОВЫ ЛОГИКИ

### 2.1. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ. КОЛИЧЕСТВО И ЕДИНИЦЫ ИЗМЕРЕНИЯ ИНФОРМАЦИИ

#### 2.1.1. ЯЗЫК КАК СПОСОБ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ

**Язык** – это система обозначений и правил для передачи сообщений. Различают языки естественные, на которых общаются люди, и искусственные (или формальные), обеспечивающие взаимодействие систем «человек – машина» или «машина – машина». К формальным языкам относятся языки программирования.

Формальный язык задается алфавитом, синтаксисом и семантикой. Теоретические основы методов проектирования языков программирования, конструирования трансляторов рассматриваются в теории формальных языков. Формальный язык – это язык, обеспечивающий удобное описание конкретных проблем, формулируемых человеком и решаемых с помощью компьютера.

На языке программирования пишется программа, позволяющая при ее выполнении компьютером (вычислительной системой) получить конкретные результаты. Язык программирования состоит из синтаксиса и семантики.

Алфавит представляет собой совокупность упорядоченных в определенном смысле символов в данном языке или системе. Эти символы называются буквами. Только символы, принадлежащие данному алфавиту, можно использовать для построения слов. Буква – это элемент алфавита. Например, алфавит языка Паскаль состоит из латинских букв (причем строчные и прописные буквы не различаются), цифр и специальных символов. Под символом понимается элемент алфавита, имеющий определенное значение. Символ, как правило, записывается в памяти компьютера восемью битами, или одним байтом.

Синтаксис (от греч. *syntaxis* – построение, порядок) – это набор правил построения слов, конструкций и структур текста в языке или системе. Некоторые авторы включают в синтаксис и алфавит. Ошибки, возникающие при написании программы и касающиеся только синтаксиса, выявляются при синтаксическом анализе, осуществляемом транслятором.

В информатике понятие «слово» имеет несколько определений. Приведем два из них. Слово – это:

- упорядоченный набор символов в заданном алфавите, имеющий определенный смысл;
- единица данных, рассматриваемая как целое при передаче и обработке данных в процессе.

**Транслятор** (от англ. *translator* – переводчик) – это программа, производящая трансляцию программы с одного языка программирования в другой.

Под семантикой (от греч. *semantikos* – обозначающий) понимается смысл каждой синтаксической конструкции в языке или системе.

В языке программирования транслятор превращает синтаксические построения в команды, понятные операционной системе и процессору. Смысловые ошибки транслятор не выявляет, их поиск осуществляет человек в процессе отладки, тестирования и даже эксплуатации программы.

Естественный язык возник очень давно. Первобытные люди могли обеспечить себе пропитание (например, добыть крупное животное) только общими усилиями. При этом они должны были общаться между собой, обмениваться информацией (скажем, предупреждать друг друга об опасности). Сначала это делалось с помощью мимики, жестов, возгласов; затем возник язык – важнейшее средство человеческого общения. Он стал средством передачи информации, значительно расширил возможности ее хра-

нения в памяти, стал одним из средств управления поведением человека. Реализуется язык в виде речи – языка в действии. В древнейшие времена говорить можно было только с находящимся рядом собеседником или с группой слушателей, так как речь исчезает в момент ее произнесения.

Речь «участвует» не только в передаче сведений от одного человека другому, но и во всех сознательных процессах, происходящих в мозгу человека. Каждый из нас мыслит с помощью речи – словами, которые не произносятся вслух. Даже когда мы молчим, мы говорим сами с собой, мыслим не только образами, но и словами. Воспринимаемые нами предметы в своем сознании мы называем словами-понятиями: стол, стул, небо, земля, утро, день, вечер, ночь. Считаем в уме мы тоже с помощью слов, обозначающих числа и вычисления: один, два, три, четыре, сложить, вычесть, умножить, разделить. Например: «Девять умножить на три равно двадцати семи», – говорим мы «в уме».

Зрительные, звуковые и другие впечатления человек запоминает в виде образов, а смысловую информацию – в виде слов. Чем раньше человек научится говорить, тем больше у него сохранится ранних детских воспоминаний.

Итак, речь представляет собой самую древнюю природную информационную технологию, которой каждый человек овладевает в первые годы жизни. Для сознательных процессов в мозгу человека эта технология является внутренней, а для передачи своих мыслей другим (с помощью речи) – внешней.

Процесс формирования речи объяснил выдающийся отечественный ученый-физиолог И.П. Павлов (1849–1936 гг.), который разработал учение о сигнальных системах. Сигнальные системы условно-рефлекторных связей формируются в коре больших полушарий головного мозга животных и человека при поступлении импульсов от внешних и внутренних раздражителей. Другими словами, мы слышим за спиной голос знакомого человека и тут же оборачиваемся. Голос в этом случае будет раздражителем, а наша реакция на обращение приятеля – реакцией на раздражитель.

**Первая сигнальная** система формируется при воздействии конкретных раздражителей – света, звука, боли, запаха и т.д. Она является формой непосредственного отражения действительности в виде ощущений.

**Вторая сигнальная** система, присущая только человеку, формируется при воздействии речевых сигналов, т.е. не непосредственных раздражителей, а их словесных обозначений. Вторая сигнальная система формируется на базе первой в процессе общения между людьми, первоначально между ребенком и родителями. Например, грудной ребенок не понимает, что именно говорит ему мать, он лишь реагирует на ее голос. Но когда он начинает понимать смысл сказанного, голос матери для него уже не просто знакомые звуки, но и самое главное средство общения. То есть первая сигнальная система, данная нам от природы, дополняется второй сигнальной системой в процессе обучения. Понятие о второй сигнальной системе ввел в 1932 г. И.П. Павлов.

### 2.1.2. ФОРМЫ ПРЕДСТАВЛЕНИЯ И ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ

При любых формах работы с информацией всегда идет речь о ее представлении в виде определенных символических структур. Наиболее распространены одномерные представления информации, при которых сообщения имеют вид последовательности символов. Так, информация представляется в письменных текстах, при передаче по каналам связи, в памяти компьютера. Однако широко используется и многомерное представление информации, причем под многомерностью понимается не только расположение элементов информации на плоскости или в пространстве в виде рисунков, схем, графов, объемных макетов и т. п., но и множественность признаков используемых символов, например: цвет, размер, вид шрифта в тексте.

Формирование представления информации называется ее кодированием. В более узком смысле под **кодированием** понимается переход от исходного представления информации, удобного для восприятия человеком, к представлению, удобному для ее хранения, передачи и обработки. В этом случае обратный переход к исходному представлению информации называется **декодированием**.

Кодирование информации осуществляется для достижения следующих целей:

- удобства физической реализации;
- удобства восприятия;
- высокой скорости передачи и обработки;
- экономичности, т.е. уменьшения избыточности сообщений;
- надежности, т.е. защиты от случайных искажений;
- сохранности, т.е. защиты от нежелательного доступа к информации.

Эти цели часто противоречат друг другу. Стремясь к экономным сообщениям, мы, тем самым, уменьшаем их надежность и удобство восприятия. Экономные сообщения могут повысить скорость обработки информации (такое сообщение будет передано или прочтено быстрее), но могут и уменьшить ее. А защита от нежелательного доступа уменьшает объем хранимой информации и замедляет работу с ней.

Рассмотрим способы представления информации в компьютере. Для записи, хранения и выдачи по запросу информации, обрабатываемой с помощью компьютера, предназначено запоминающее устройство (или память) вычислительной машины. В отличие от обычной словесной формы, принятой в письменной речи, информация в памяти компьютера записывается в форме цифрового двоичного кода. Это объясняется тем, что электронные элементы, на которых строится память компьютера, находятся только в одном из двух устойчивых состояний – их можно интерпретировать как **0** или **1**.

Количество информации, которое может помещаться в один элемент памяти (0 или 1), называется **битом**. Однако если соединить несколько таких элементов в ячейку, то можно сохранить в запоминающем устройстве столько информации, сколько требуется. Последовательность бит, рассматриваемых аппаратной частью компьютера как единое целое, называется **машинным словом**. Слово «бит» происходит от английских слов **binary** – двойной и **digit** – цифра:

$$\text{Binary} + \text{digit} = \text{BIT}.$$

Так как оперативная память компьютера состоит из конечной последовательности слов, а слова – из конечной последовательности бит, то объем представляемой в компьютере информации ограничен емкостью памяти, а числовая информация может быть представлена только с определенной точностью, зависящей от архитектуры памяти данной компьютера.

### 2.1.3. КОЛИЧЕСТВО И ЕДИНИЦЫ ИЗМЕРЕНИЯ ИНФОРМАЦИИ

Для теоретической информатики информация играет такую же роль, как и вещество в физике. И подобно тому, как веществу можно приписать довольно большое количество характеристик (массу, заряд, объем и т.д.), так и для информации имеется пусть и не столь большой, но достаточно представительный набор характеристик. Для характеристики информации (как и вещества) и имеются единицы измерения, что позволяет некоторой порции информации приписывать числа – количественные характеристики информации.

На сегодняшний день наиболее известны объемный, энтропийный и алгоритмический способы измерения информации.

**Объемный** – самый простой и грубый способ измерения информации. Соответствующую количественную оценку информации естественно назвать объемом информации. Объем информации в сообщении – это количество символов в нем. Поскольку одно и то же число может быть записано разными способами, т.е. с использованием

разных алфавитов (например: двадцать один; 21; 10101; XXI), объемный способ чувствителен к форме представления (записи) сообщения. Повторим: в вычислительной технике вся обрабатываемая и хранимая информация независимо от ее природы (число, текст, изображение) представлена в двоичной форме. Такая стандартизация позволила ввести две стандартные единицы измерения: *бит* и *байт*.

Название «*байт*» (слово *byte* представляет собой сокращение словосочетания *BinarY TErm* – «двоичный терм») было впервые использовано в 1956 году В. Бухгольцем при проектировании первого суперкомпьютера IBM 7030 для пучка одновременно передаваемых в устройствах ввода-вывода шести битов. Позже, в рамках того же проекта, байт был расширен до восьми бит.

Ряд ЭВМ 1950-х и 1960-х годов (БЭСМ-6, М-220) использовали 6-битовые символы в 48-битовых или 60-битовых машинных словах. В некоторых моделях ЭВМ производства Burroughs Computer Corporation (ныне Unisys) размер символа был равен 9 битам. В советской ЭВМ Минск-32 использовался 7-битный байт.

Байтовая адресация памяти была впервые применена в системе IBM System/360. В более ранних компьютерах адресовать можно было только целиком машинное слово, состоявшее из нескольких байт, что затрудняло обработку текстовых данных.

Восьмибитные байты были приняты в System/360, вероятно, из-за использования BCD-формата представления чисел: одна десятичная цифра (0–9) требует 4 бита (тетраду) для хранения; один 8-битный байт может представлять две десятичные цифры. 6-битные байты могут хранить только по одной десятичной цифре, два бита остаются незадействованными.

По другой версии, 8-битный размер байта связан с 8-битным же числовым представлением символов в кодировке EBCDIC.

Постепенно 8-битные байты стали стандартом де-факто и с начала 1970-х в большинстве компьютеров байты состоят из 8 бит и размер машинного слова кратен 8 битам. Из соображений удобства единицы нетекстовых типов данных также делают кратными 8 битам, например:

- размер одного сэмпла в звуковых файлах равен 8, 16 или 24 битам
- размер пикселя в системе RGB равен 24 битам (по 8 бит на цвет)

В теории информации и кодирования принят *энтропийный* способ измерения информации. Он исходит из следующей модели. Получатель информации (сообщения) имеет определенные представления о возможных наступлениях некоторых событий. Эти представления в общем случае недостоверны и выражаются вероятностями, с которыми он ожидает то или иное событие.

Общая мера неопределенности (энтропия) характеризуется некоторой математической зависимостью от совокупности этих вероятностей. Количество информации в сообщении определяется тем, насколько уменьшится эта мера после получения сообщения. Поясним эту идею на примере. Пусть имеется колода из 32 различных карт. Возможность выбора одной карты из колоды равна 32. Априори (доопытно, до того, как произведен выбор) естественно предположить, что наши шансы выбрать некоторую определенную карту одинаковы для всех карт колоды. Произведя выбор, мы устраняем эту априорную неопределенность.

Нашу априорную неопределенность можно было бы охарактеризовать количеством возможных равновероятных выборов. Если теперь определить количество информации как меру устраненной неопределенности, то и полученную в результате выбора информацию можно охарактеризовать числом 32.

Однако в теории информации получила использование другая количественная оценка, а именно – логарифм от описанной выше оценки по основанию 2:

$$H = \log_2 m, \quad (2.1)$$

где  $m$  – число возможных равновероятных выборов (при  $m = 2$ ,  $H = 1$ ).

То есть для выбора из колоды имеем следующую оценку количества информации, получаемую в результате выбора:

$$H = \log_2 32 = 5.$$

Полученная оценка имеет интересную интерпретацию. Она характеризует число двоичных вопросов, ответы на которые позволяют выбрать либо «да», либо «нет». Для выбора дамы пик такими вопросами и ответами будут:

Вопрос	Ответ
Карта красной масти?	«Нет» – 0
Трефы?	«Нет» – 0
Одна из четырех старших?	«Да» – 1
Одна из двух старших?	«Нет» – 0
Дама?	«Да» – 1

Этот выбор можно описать последовательностью из пяти двоичных символов:

$$00101 \text{ (0 – «нет», 1 – «да»)}.$$

На первый взгляд, может показаться, что эта интерпретация не годится в случае, когда количество выборов не равно степени двойки, так как получается нецелое количество вопросов. Например, если взять колоду из 36 карт (добавлены шестерки), то можно заметить, что для того, чтобы выяснить у участника «эксперимента», какую карту он выбрал, в ряде случаев понадобится пять вопросов (как и в рассмотренном случае), а в ряде случаев – шесть. Усреднение по случаям и дает получаемую по формуле (2.1) нецелую величину.

К. Шеннону принадлежит обобщение  $H$  на случай, когда  $H$  зависит не только от  $m$ , но и от вероятностей возможных выборов (для сообщения – вероятности выбора символов). Для количества собственной индивидуальной информации он предложил соотношение:

$$h_i = \log \left( \frac{1}{P_i} \right) = -\log P_i \quad (2.2)$$

где  $P_i$  – вероятность выбора  $i$ -го символа алфавита.

Более удобно пользоваться средним значением количества информации, приходящимся на один символ алфавита:

$$H = -\sum_{i=1}^m P_i \log P_i \quad (2.3)$$

При равновероятных выборах все  $P_i = 1/m$ , и получается прежняя формула:

$$H = \log_2 m.$$

В алгоритмической теории информации (в разделе теории алгоритмов) предлагается *алгоритмический* способ оценки информации в сообщении. Этот способ кратко можно охарактеризовать следующим:

1) каждый согласится, что слово 0101...01 сложнее слова 00...0, а слово, где 0 и 1 выбираются из эксперимента – бросания монеты (где 0 – герб, 1 – число), сложнее предыдущих;

2) компьютерная программа, производящая слово из одних нулей, крайне проста: печатать один и тот же символ. Для получения 0101...01 нужна чуть более сложная программа, печатающая символ, противоположный только что напечатанному;

3) случайная, не обладающая никакими закономерностями последовательность не может быть произведена никакой «короткой» программой. Длина программы, производящей хаотичную последовательность, должна быть близка к длине последней.

Приведенные рассуждения позволяют предположить, что любому сообщению можно приписать количественную характеристику, отражающую сложность (размер) программы, которая позволяет ее произвести.

Так как имеется много разных вычислительных машин и разных языков программирования (разных способов задания алгоритма), то для определенности используется конкретная вычислительная машина, а сложность слова (сообщения) определяется как минимальное число внутренних состояний машины, требующееся для его воспроизведе-

дения. В алгоритмической теории информации применяются и другие способы задания сложности.

Кроме бита и байта, для измерения количества информации используются также более крупные единицы:

- 1 килобайт = 1024 байт ( $2^{10}$  байт);
- 1 мегабайт = 1024 Кбайт ( $2^{20}$  байт);
- 1 гигабайт = 1024 Мбайт ( $2^{30}$  байт);
- 1 терабайт = 1024 Гбайт ( $2^{40}$  байт);
- 1 петабайт = 1024 Тбайт ( $2^{50}$  байт);
- 1 эксабайт = 1024 Пбайт ( $2^{60}$  байт).

## 2.2. СИСТЕМЫ СЧИСЛЕНИЯ, ИСПОЛЬЗУЕМЫЕ В КОМПЬЮТЕРЕ

### 2.2.1. ПОНЯТИЕ СИСТЕМЫ СЧИСЛЕНИЯ

*Системой счисления* называется совокупность приемов наименования и записи чисел. В любой системе счисления для представления чисел выбираются некоторые символы (слова или знаки), называемые базисными числами, а все остальные числа получаются в результате каких-либо операций из базисных чисел данной системы счисления. Символы, используемые для записи чисел, могут быть любыми, только они должны быть разными и значение каждого из них должно быть известно. В современном мире наиболее распространенным является представление чисел посредством арабских цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 – специальных знаков, используемых для записи чисел.

Системы счисления различаются выбором базисных чисел и правилами образования из них остальных чисел. Например, в римской системе счисления базисными являются числа 1, 5, 10, 50, 100, 500, 1000, которые обозначаются знаками, соответственно I, V, X, L, C, D, M, а другие числа получаются путем сложения и вычитания базисных: если цифра справа меньше или равна цифре слева, то эти цифры складываются; если цифра слева меньше, чем цифра справа, то левая цифра вычитается из правой. Так, например, число 146 в римской системе счисления имеет вид: CXLVI (C – 100, XL – 40, VI – 6). Здесь число 40 получается посредством вычитания из 50 числа 10, а 6 – посредством сложения чисел 5 и 1.

Системы счисления, в которых любое число получается путем сложения или вычитания базисных чисел, называются *аддитивными*. При таком представлении чисел правила сложения для небольших чисел очевидны и просты, однако если возникает необходимость выполнять операции сложения над большими числами или операции умножения и деления, то римская система счисления оказывается неудобной. В этой ситуации преимущество имеют позиционные системы счисления, хотя в них, как правило, представления чисел далеко не так просты и очевидны, как в римской.

Систематичность представления, основанная на «позиционном весе» цифр, обеспечивает простоту выполнения операций умножения и деления.

В римской системе счисления каждый числовой знак в записи любого числа имеет одно и то же значение, т.е. значение числового знака не зависит от его расположения и записи числа. Таким образом, римская система счисления не является позиционной.

### 2.2.2. ПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ

Для изображения (или представления) чисел в настоящее время используются в основном позиционные системы счисления. Привычной для всех является десятичная система счисления. В этой системе для записи любых чисел используется только десять разных знаков (цифр): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Эти цифры введены для обозначения первых десяти последовательных чисел, а следующие числа (начиная с 10 и т.д.) обо-

значаются уже без использования новых знаков (цифр). Таким образом, сделан важный шаг в построении системы счисления: значение каждой цифры поставлено в зависимость от того места, где она стоит в изображении числа.

Система называется *позиционной*, если значение каждой цифры (ее вес) изменяется в зависимости от ее положения (позиции) в последовательности цифр, изображающих число. Если это условие не выполняется, то система счисления является непозиционной (например, римская).

Десятичная позиционная система счисления основана на том, что десять единиц каждого разряда объединяются в одну единицу соседнего старшего разряда. Таким образом, каждый разряд имеет вес, равный степени 10. Например, в зависимости числа 343.32 цифра 3 повторена три раза, при этом самая левая цифра 3 означает количество сотен (ее вес равен  $10^2$ ); цифра 3, стоящая перед точкой, означает количество единиц (ее вес равен 100), а самая правая цифра 3 (после точки) – количество десятых долей единицы (ее вес равен  $10^{-1}$ ), так что последовательность цифр 343.32 представляет собой сокращенную запись выражения:

$$3 \times 10^2 + 4 \times 10^1 + 3 \times 10^0 + 3 \times 10^{-1} + 2 \times 10^{-2}.$$

Десятичная запись любого числа  $X$  в виде последовательности цифр:

$$a_n a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m} \dots \quad (2.4)$$

основана на представлении этого числа в виде полинома:

$$X = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \dots + a_1 \times 10^1 + a_0 \times 10^0 + \\ + a_{-1} \times 10^{-1} + \dots + a_{-m} \times 10^{-m} + \dots \quad (2.5)$$

где каждый коэффициент  $a_i$  может быть одним из чисел, для обозначения которых введены специальные знаки.

Запись числа  $X$  в виде последовательности (2.4) представляет собой просто перечисление всех коэффициентов этого полинома. Точка, отделяющая целую часть числа от дробной, служит для фиксации конкретных значений каждой позиции в этой последовательности цифр и является началом отсчета.

Число  $K$  единица какого-либо разряда, объединяемых в единицу более старшего разряда, называется основанием позиционной системы, а сама система счисления называется  $K$ -ичной. Например, основанием десятичной системы счисления является число 10; двоичной – число 2; троичной – число 3 и т.д. Для записи произвольного числа в  $K$ -ичной системе счисления достаточно иметь  $K$  разных цифр  $a_i$ ,  $i = 1, K$ . Например, в троичной системе счисления любое число может быть выражено посредством цифр 0, 1, 2. Эти цифры служат для обозначения некоторых различных целых чисел, называемых базисными.

Числа можно записать как суммы степеней не только числа 10, но и любого другого натурального числа, большего единицы. Например, в Древнем Вавилоне использовалась система счисления с основанием 60. Деление часа на 60 минут, а минуты – на 60 секунд заимствовано именно из этой системы счисления. А то, что человечество выбрало в качестве основания системы счисления число 10, вероятно, связано с тем, что природа наделила людей десятью пальцами.

Запись произвольного числа  $X$  в  $K$ -ичной позиционной системе счисления основывается на представлении этого числа в виде полинома:

$$X = a_n \times K^n + a_{n-1} \times K^{n-1} + \dots + a_1 \times K^1 + a_0 \times K^0 + \\ + a_{-1} \times K^{-1} + \dots + a_{-m} \times K^{-m} + \dots \quad (2.6)$$

где каждый коэффициент  $a_i$  может быть одним из базисных чисел и изображается одной цифрой.

Как и в десятичной системе счисления, число  $X$ , представленное в  $K$ -ичной системе счисления, можно кратко записать в виде последовательности (2.4) путем перечисления всех коэффициентов полинома (2.6) с указанием позиционной точки. После-

довательность цифр, стоящая в (2.4), является изображением числа  $X$  в  $K$ -ичной системе счисления. Базисные числа должны быть выбраны так, чтобы любое число  $X$  могло быть представлено в виде полинома (2.6). Обычно в качестве базисных чисел берутся последовательные целые числа от 0 до  $K - 1$  включительно.

Арифметические действия над числами в любой позиционной системе счисления производятся по тем же принципам, что и в десятичной системе, так как все они основываются на правилах выполнения действий над соответствующими полиномами. При этом нужно только пользоваться теми таблицами сложения и умножения, которые имеют место при данном основании  $K$  системы счисления.

Для указания того, в какой системе счисления записано число, условимся при его изображении основание системы счисления указывать в виде нижнего индекса при нем, например:  $35,64_8$ .

### 2.2.3. ДВОИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ

В современной вычислительной технике, в устройствах автоматики и связи широко применяется двоичная система счисления. Это система счисления с наименьшим возможным основанием. В ней для изображения числа используются только две цифры: 0 и 1.

Произвольное число  $X$  в двоичной системе представляется в виде полинома:

$$X = a_n \times 2^n + a_{n-1} \times 2^{n-1} + \dots + a_1 \times 2^1 + a_0 \times 2^0 + \\ + a_{-1} \times 2^{-1} + \dots + a_{-m} \times 2^{-m} + \dots \quad (2.7)$$

где каждый коэффициент  $a_i$  может быть либо 0, либо 1.

Примеры изображения чисел в двоичной системе счисления:

$1 = 1_2$	$5 = 101_2$	$9 = 1001_2$
$2 = 10_2$	$6 = 110_2$	$10 = 1010_2$
$3 = 11_2$	$7 = 111_2$	$0,5 = 0,1_2$
$4 = 100_2$	$8 = 1000_2$	$0,25 = 0,01_2$

Таблица сложения чисел в двоичной системе счисления:

$$\begin{aligned} 0 + 0 &= 0 \\ 1 + 0 &= 1 \\ 0 + 1 &= 1 \\ 1 + 1 &= 10 \end{aligned}$$

Таблица умножения чисел в двоичной системе счисления:

$$\begin{aligned} 0 \times 0 &= 0 \\ 1 \times 0 &= 0 \\ 0 \times 1 &= 0 \\ 1 \times 1 &= 1 \end{aligned}$$

Так как в двоичной системе счисления для изображения любых чисел используются только две цифры, то при построении компьютера можно применять лишь элементы, которые могут находиться только в двух состояниях (например, высокое или низкое напряжение в цепи тока, наличие или отсутствие электрического импульса и т.п.). Это обстоятельство, а также простота выполнения арифметических операций являются причиной того, что в большинстве современных компьютеров используется двоичная система счисления.

### 2.2.4. ПРАВИЛА ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ В ДВОИЧНОЙ СИСТЕМЕ СЧИСЛЕНИЯ

**Сложение.** Операция сложения выполняется так же, как и в десятичной системе счисления. Переполнение разряда приводит к появлению единицы в следующем разряде.

$$\begin{aligned} 0 + 0 &= 0; \\ 0 + 1 &= 1; \\ 1 + 1 &= 10; \end{aligned}$$

**Вычитание.** Поскольку большинство современных компьютеров располагает только одним аппаратным сумматором, с помощью которого реализуются арифметические операции, вычитание сводится к сложению с отрицательным числом:

$$15 - 8 = 15 + (-8).$$

**Правила вычитания в двоичной системе счисления.** Операции вычитания путем сложения дополнительных кодов выполняются по следующему алгоритму:

- 1) преобразовать отрицательное число из формы со знаком в дополнительный код;
- 2) выполнить операцию двоичного сложения над всеми разрядами, включая знаковый, игнорируя единицу переноса из самого высокого разряда;
- 3) если знаковый разряд суммы равен единице (что означает получение отрицательного результата в форме дополнительного кода), необходимо перевести результат в знаковую форму, используя второе свойство дополнений.

Таким образом, при выполнении операций сложения и вычитания арифметико-логическому устройству процессора приходится выполнять поразрядное сложение с переносом, инвертирование и проверку на знак двоичных чисел.

В тех случаях, когда необходимо произвести арифметические действия над числами больше 127, они размещаются уже не в одном, а в двух и более регистрах.

**Умножение.** Если наряду с перечисленными операциями выполнить операции сдвига, то с помощью сумматора можно выполнить и умножение, которое сводится к серии повторных сложений. Если цифра в нулевой позиции множителя равна единице, то множимое переписывается под соответствующими разрядами, умножение на последующие единицы приводит к сдвигу слагаемого влево на одну позицию. Если цифра множителя равна 0, то следующее слагаемое смещается на две позиции влево.

**Деление.** При выполнении операции деления несколько раз производится операция вычитания. Поэтому предварительно следует найти дополнительный код делителя. Деление выполняется путем повторного вычитания и сдвига.

### 2.2.5. ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ

При решении задач с помощью компьютера исходные данные обычно задаются в десятичной системе счисления; в этой же системе, как правило, нужно получить и окончательные результаты. Так как в современных компьютерах данные кодируются в основном в двоичных кодах, то, в частности, возникает необходимость перевода чисел из десятичной в двоичную систему счисления и наоборот.

**Представление в компьютере отрицательных чисел.** Следует иметь в виду, что в памяти компьютера двоичные числа хранятся в регистрах, состоящих из восьми ячеек, т.е. минимальное двоичное число, которое можно разместить в памяти, должно быть восьмиразрядным. При этом в незаполненные ячейки регистра (в старших разрядах) записываются нули.

В отличие от десятичной, в двоичной системе счисления отсутствуют специальные символы, обозначающие знак числа – положительный («+») или отрицательный («-»), поэтому для представления двоичных отрицательных чисел используется следующее правило:

- 1) форма значения со знаком: старший (левый) разряд метится как знаковый и содержит информацию только о знаке числа:

1 – число отрицательное; 0 – число положительное;

- 2) остальные разряды отводятся под абсолютную величину числа:

$$5_{10} = 0000\ 0101_2$$

$$-5_{10} = 1000\ 0101_2$$

Компьютеры обычно строятся таким образом, чтобы отрицательные числа были представлены в дополнительном коде, поскольку это дает существенную экономию времени при выполнении с ними арифметических операций.

### 2.2.6. ДРУГИЕ ПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ

Неудобство использования двоичной системы счисления заключается в громоздкости записи чисел. Это неудобство не имеет существенного значения для компьютера. Однако если возникает необходимость кодировать информацию «вручную», например, при составлении программы на машинном языке, то предпочтительнее пользоваться восьмеричной или шестнадцатеричной системами счисления (в силу их свойств, которые будут рассмотрены далее).

В восьмеричной системе счисления базисными являются числа 0, 1, 2, 3, 4, 5, 6, 7. Запись любого числа в этой системе основывается на его разложении по степеням числа 8 с коэффициентами, которыми являются указанные базисные числа.

Например, десятичное число 83,5 в восьмеричной системе будет изображаться в виде 123,4. Действительно, эта запись, по определению, означает представление числа в виде полинома:

$$1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 + 4 \times 8^{-1} = 64 + 16 + 3 + 4/8 = 83,5.$$

В шестнадцатеричной системе счисления базисными являются числа от 0 до 15. Эта система отличается от рассмотренных тем, что в ней не хватает общепринятых (арабских) цифр для обозначения всех базисных чисел, поэтому приходится вводить в употребление новые символы. Обычно для обозначения первых десяти целых чисел (от 0 до 9) используются арабские цифры, а для следующих целых чисел (от 10 до 15) – буквенные обозначения из латинского алфавита: A, B, C, D, E, F.

Например, десятичное число 175,5 в шестнадцатеричной системе записывается как AF,816. Действительно:

$$10 \times 16^1 + 15 \times 16^0 + 8 \times 16^{-1} = 160 + 15 + 8/16 = 175,5.$$

Каждая тройка двоичных разрядов соответствует одной восьмеричной цифре, каждая четверка – шестнадцатеричной. Отсюда следует простота преобразований из двоичной системы счисления в восьмеричную и шестнадцатеричную, например:

$$11010011_2 = 1101\ 0011_2 = D3_{16};$$

$$11010011_2 = 011\ 010\ 011_2 = 323_8.$$

Если исходное количество бит не кратно 3 или 4, добавляются нули слева.

Обратное преобразование аналогично:

$$B9_{16} = 1011\ 1001_2;$$

$$270_8 = 10\ 111\ 000_2.$$

Перевод из десятичной системы счисления в K-ичную производится аналогично переводу в двоичную систему путем целочисленного деления десятичного числа на основание системы K до тех пор, пока частное не станет меньше основания.

Перевод из m-ичной системы в десятичную производится путем сложения произведений соответствующего десятичного эквивалента символа числа в m-ичной системе на вес i-го знакоместа.

Пример перевода из шестнадцатеричной системы счисления в десятичную:

$$15B_{16} = 1 \times 16^2 + 5 \times 16^1 + 11 \times 16^0 = 256 + 80 + 11 = 347_{10}.$$

## 2.3. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В ПАМЯТИ КОМПЬЮТЕРА

### 2.3.1. ПРЕДСТАВЛЕНИЕ ЦЕЛЫХ ЧИСЕЛ БЕЗ ЗНАКА И СО ЗНАКОМ

Введем для простоты сначала основные понятия на примере четырехбитовых машинных слов. Такой размер слова обеспечивает хранение десятичных чисел только от 0 до 15 и поэтому не представляет практического значения. Однако такой пример менее

громоздкий, а основные закономерности представления этих чисел сохраняют силу для машинного слова любого размера.

Предположим, что процессор компьютера способен увеличивать (прибавлять 1) и дополнять (инвертировать) четырехбитовые слова. Например, результатом увеличения слова 1100 является 1101, а результатом дополнения этого слова является 0011. Рассмотрим слово 0000, представляющее десятичное число 0. В результате увеличения содержимое этого слова станет равным 0001, что соответствует десятичному числу 1. Продолжая последовательно увеличивать четырехбитовые слова, приходим к ситуации, когда, увеличивая слово 1111 (которое представляет десятичное число 15), получим в результате слово 0000, т.е.

$$1111 + 1 = 0000 \quad (15 + 1 = 0);$$

при этом получили неверную арифметическую операцию и вернулись в исходное состояние. Это произошло из-за того, что слово памяти может состоять только из конечного числа бит. Таким образом, числовая система компьютера является *конечной* и *циклической*.

Такой ситуации, приводящей к неверному арифметическому результату, можно избежать, если битовую конфигурацию 1111 принять за код для -1. Тогда 1110 интерпретируется как -2; 1101 – как -3 и т.д. до 1000 – как -8. Тем самым получили другую числовую систему – со знаком, содержащую как положительные, так и отрицательные числа. В этой системе половина четырехбитовых конфигураций, начинающаяся с единицы, интерпретируется как отрицательные числа, а другая половина, начинающаяся с нуля, – как положительные числа или нуль. Поэтому старший бит числа (третий по счету, если нумерацию бит начинать с нуля справа налево) называется *знаковым битом*.

Числовая система со знаком также конечна и циклическа, однако в этом случае арифметически неверный результат даст попытка увеличить число 8 на единицу. Преимущество введения числовой системы со знаком заключается в возможности представления как положительных, так и отрицательных чисел.

Если знаковый бит равен нулю, то значение числа легко вычисляется – игнорируется знаковый бит, а оставшиеся три бита интерпретируются как двоичный код десятичного числа. Например, слово 0110 представляет двоичное число 110, которое равно десятичному числу 6.

Для оценки отрицательного числа нужно изменить его знак. Рассмотрим четырехбитовое число  $K$  в системе со знаком. Тогда

$$-K = (-1 - k) + 1,$$

следовательно, для вычисления значения  $-k$  необходимо вычесть  $K$  из -1 (т.е. из 1111) и затем прибавить единицу (т.е. 0001). Заметим, что операция вычитания всегда возможна, никогда не требует заёма и равнозначна операции инвертирования битов вычитаемого. Например:

$$1111 - 1011 = 0100;$$

здесь в вычитаемом, равном 1011, единицы перешли в нули, а нуль – в единицу.

Инвертирование бит в слове называется *дополнением до единицы*. Для определения отрицательного значения числа  $K$  надо к его дополнению до единицы прибавить единицу (согласно ранее приведенному равенству). Инвертирование бит в слове с добавлением единицы к младшему биту называется *дополнением до двух*. Например, требуется найти, какое число закодировано в слове 1001. Для этого сначала выполняем операцию инвертирования:  $1001 - 0110$ , а затем к полученному результату прибавляем единицу:  $0110 + 1 = 0111$ , получаем число, которое является дополнительным кодом числа 7. Таким образом, значением 1001 является отрицательное число 7, т.е. -7.

### 2.3.2. ИНДИКАТОРЫ ПЕРЕНОСА И ПЕРЕПОЛНЕНИЯ

Рассмотрим более подробно ситуацию, приводящую при увеличении четырехбитового числа (т. е. при прибавлении к нему единицы) к неверному арифметическому результату и возникающую из-за конечности числовой системы компьютера. В числовой системе без знака такая проблема встает при увеличении слова 1111, при этом имеет место перенос единицы из знакового бита. В случае системы чисел со знаком перенос из старшего бита дает верный результат:

$$1111 + 0001 = 0000$$

(что правильно:  $-1 + +1 = 0$ ), но увеличение слова 0111 приводит к ошибочной ситуации:

$$0111 + 1 = 1000 \quad (7 + 1 = -8),$$

при этом имеет место **перенос в знаковый бит**.

В процессоре компьютера (устройстве, в котором выполняются арифметические операции) имеется два индикатора – **индикатор переноса** и **индикатор переполнения**. Каждый индикатор содержит один бит информации и может быть процессором установлен (в этом случае ему придается значение, равное единице) или сброшен (равен нулю). Индикатор переноса указывает на перенос из знакового бита, а индикатор переполнения – на перенос в знаковый бит. Таким образом, после завершения операции, в которой происходит перенос в старший бит, процессор устанавливает индикатор переполнения; если такого переноса нет, то индикатор переполнения сбрасывается. Индикатор переноса обрабатывается аналогичным образом.

Важно то, что после выполнения операции процессор компьютера сигнализирует о состоянии индикаторов, и их можно проверить. Если состояние индикаторов указывает на неправильный арифметический результат, то необходимо исправить эту ситуацию, т.е. программе предоставляется возможность контролировать правильность выполнения арифметических операций.

#### Пример 1

В результате сложение двух чисел

$$0101 + 0011 = 1000$$

произошел перенос в знаковый бит, а переноса из знакового бита не было. Таким образом, после завершения этой операции индикатор переноса будет сброшен, а индикатор переполнения установлен. Поэтому если рассматривать эти два числа как целые без знака, то результат является арифметически правильным, так как индикатор переноса сброшен. Если же рассматривать числа в системе со знаком, то бит переполнения показывает, что произошло изменение знака (перенос в знаковый бит есть, а переноса из него – нет), поэтому арифметический результат неправильный.

#### Пример 2

В результате сложения чисел

$$1101 + 0101 = 0010$$

происходит перенос в знаковый бит и из знакового бита. Поэтому будет установлен индикатор переноса, а индикатор переполнения сброшен. Следовательно, в системе чисел без знака результат является арифметически неправильным, а в системе чисел со знаком – правильным.

#### Пример 3

В результате выполнения операции вычитания:

$$1001 - 0011 = 1001 + (-0011) = 1001 + (1100 + 1) = 1001 + 1101 = 0110$$

происходит перенос из знакового бита, а переноса в знаковый бит нет. Следовательно, индикатор переноса будет сброшен, а индикатор переполнения установлен; это указывает на то, что в данном примере в системе без знака результат арифметически правильный, а в системе со знаком – неправильный.

### 2.3.3. ПРЕДСТАВЛЕНИЕ СИМВОЛЬНОЙ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ

В отличие от обычной словесной формы, принятой в письменном виде, символьная информация хранится и обрабатывается в памяти компьютера в форме цифрового кода. Например, можно обозначить каждую букву числами, соответствующими ее порядковому номеру в алфавите: А – 01, Б – 02, В – 03, Г – 04, Э – 30, Ю – 31, Я – 32. Точно так же можно договориться обозначать точку числом 33, запятую – числом 34 и т.д. Так как в устройствах автоматической обработки информации используются двоичные коды, обозначения букв надо перевести в двоичную систему. Тогда буквы будут обозначаться следующим образом: А – 000001, Б – 000010, В – 000011, Г – 000100, Э – 011110, Ю – 011111, Я – 100000. При таком кодировании любое слово можно представить в виде последовательности кодовых групп, составленных из 0 и 1. Например, слово «ЭВМ» выглядит так: 011110 000011 001110.

При преобразовании символов (знаков) в цифровой код между множествами символов и кодов должно быть взаимно однозначное соответствие, т.е. разным символам должны быть назначены разные цифровые коды, и наоборот. Это условие является единственным необходимым требованием при построении схемы преобразования символов в числа. Однако существует ряд практических соглашений, принимаемых при построении схемы преобразования из соображений наглядности, эффективности, стандартизации. Например, какое бы число ни назначили коду для знака 0 (не следует путать с числом 0), знаку 1 удобно назначить число, на единицу большее, чем код 0, и т.д. до знака 9. Аналогичная ситуация возникает и при кодировке букв алфавита: код для Б на единицу больше кода для А, а код для В – на единицу больше кода для Б и т.д. Таким образом, для наглядности и легкости запоминания целесообразно множества символов, упорядоченных по какому-либо признаку (например, лексикографическому), кодировать также с помощью упорядоченной последовательности чисел.

Другой важный момент при организации кодировки символьной информации – эффективное использование оперативной памяти компьютера. Так как общепотребительными являются примерно 100 знаков (сюда, помимо цифр, букв русского и английского алфавитов, знаков препинания, арифметических знаков, входят знаки перевода строки, возврата каретки, возврата на шаг и т. п.), то для взаимно однозначного преобразования всех знаков в коды достаточно примерно сотни чисел. Важность этого выбора заключается в том, что для размещения числа из этого диапазона в оперативной памяти достаточно одного байта, а не машинного слова. Следовательно, при такой организации кодировки достигается существенная экономия объема памяти.

При назначении кодов знакам надо также учитывать соглашения, касающиеся стандартизации кодировки. Можно назначить знаковые коды по своему выбору, но при этом возникнут трудности, связанные с необходимостью обмена информацией с другими организациями, использующими кодировку, отличную от нашей. В настоящее время существует несколько широко распространенных схем кодирования. Например, двоично-десятичный код **BCD** (Binary-Coded Decimal) используется для представления чисел, при котором каждая десятичная цифра записывается своим четырехбитовым двоичным эквивалентом.

Этот код может оказаться полезным, когда нужно преобразовать строку числовых знаков, например строку из числовых знаков 2537 в число 2537, над которым затем будут производиться арифметические действия. Расширением кода BCD является **EBCDIC** (Extended Binary-Coded Decimal Interchange Code) – расширенный двоично-десятичный код обмена информацией, который преобразует как числовые, так и буквенные строки.

В компьютере применяется код **ASCII** (American Standard Code for Information Interchange) – Американский стандартный код обмена информацией. Этот код генериру-

ется некоторыми внешними устройствами – принтером, алфавитно-цифровым печатающим устройством (АЦПУ) – и используется для обмена данными между ними и оперативной памятью компьютера. Например, когда нажимаем на терминале клавишу <G>, то в результате этого действия код ASCII для символа G (1000111) передается в компьютер. А если надо этот символ распечатать на АЦПУ, то его код ASCII должен быть послан на печатающее устройство.

Отечественной версией кода ASCII является код **КОИ-7** (двоичный семибитовый код обмена информацией), который совпадает с ним, за исключением букв русского алфавита.

#### 2.3.4. ФОРМАТЫ ДАННЫХ

Под данными будем понимать информацию, представленную в виде, пригодном для обработки автоматическими средствами, например в двоичном коде. Формат представления данных в памяти компьютера зависит от ее архитектуры.

Данные, обрабатываемые компьютером, подразделяют на три группы: логические коды, числа с фиксированной точкой и числа с плавающей точкой.

**Представление логических кодов.** Логические коды могут размещаться в отдельных байтах и в словах. Для их представления используются все разряды: для байта – от 0-го до 7-го, для слова – от 0-го до 15-го. Логическими кодами могут быть представлены символьные величины, числа без знака и битовые величины.

Символьные величины задаются в коде ASCII (КОИ-7), каждый символ занимает один байт, разряд 7 которого всегда содержит 0. Символы строки размещаются в последовательно адресуемых байтах оперативной памяти. Например, символьная строка ABCDE (коды ASCII: A – 1018, B – 1028, C – 1038, D – 1048, E – 1058), первый знак которой помещается в ячейку с адресом 1000 (адреса представлены в восьмеричной системе счисления), размещается в оперативной памяти следующим образом: числа без знака для байта имеют диапазон представления от 000 до 3778, для слова – от 000000 до 1777778; битовые величины задают значения отдельных разрядов байта или слова.

**Представление чисел в формате с фиксированной точкой.** Фиксированной называется такая точка, позиция которой фиксируется форматом представления чисел. Это такой способ разделения целой и дробной частей числа, при котором положение точки не меняется в ходе выполнения операций над данным числом. Числа, представленные в формате с фиксированной точкой, могут занимать один байт или слово. Если число с фиксированной точкой занимает один байт, то для его представления используются разряды с 0-го по 6-й. Разряд 7 называется знаковым. При размещении числа с фиксированной точкой в слове для его представления используются разряды с 0-го по 14-й. Знак числа содержится в разряде 15. Значения знакового разряда: 0 – для положительных чисел; 1 – для отрицательных чисел.

Отрицательные числа в формате с фиксированной точкой представляются в дополнительном коде (посредством операции дополнения до двух).

Диапазон представления чисел в формате с фиксированной точкой: для байта – от  $-128_{10}$  до  $+127_{10}$ ; для слова – от  $-32768_{10}$  до  $+32767_{10}$ . При выполнении операций над числами, представленными в формате с фиксированной точкой, они масштабируются таким образом, чтобы каждое число лежало в интервале  $(-1, +1)$ . Другими словами, в этом случае компьютер оперирует только с числами, по модулю не превосходящими единицу. При этом необходимо следить за тем, чтобы результат не получился большим, чем  $2^k - 1$ , где  $k$  – число разрядов, отведенных для представления в машине. Такая опасность есть при выполнении операций сложения и деления.

Опасность представляют также операции вычитания и умножения. При вычитании может получиться так, что разность станет числом, меньшим, чем представляется в

машине, и результат исчезнет. При многократном умножении (из-за того, что умножаются числа, меньшие единицы) может произойти то же самое. Поэтому при использовании формата представления чисел с фиксированной точкой приходится следить как за случаями возможного переполнения разрядной сетки машины, так и за случаями, связанными с появлением машинного нуля.

Необходимость постоянно следить за тем, чтобы числа в машине не вышли за пределы интервала  $(-1, +1)$ , а также неизбежное в таких устройствах накопление абсолютной погрешности вычислений из-за перемасштабирования, при котором цифры младших разрядов (а именно в них накапливается абсолютная погрешность) передвигаются в старшие разряды, привели к тому, что в универсальных ЭВМ представление чисел в формате с фиксированной точкой практически перестало применяться. Оно сохраняется в специализированных вычислительных системах, где диапазон изменения чисел заранее проанализирован, в некоторых микропроцессорах и микроЭВМ.

**Представление чисел в формате с плавающей точкой.** Плавающей называется такая точка, позиция которой не фиксируется форматом числа. Любое вещественное число  $X$ , представленное в системе счисления с основанием  $N$ , можно записать в виде

$$X = \pm m N^{\pm p},$$

где  $m$  – мантисса,  $p$  – характеристика (или порядок) числа.

Если  $|m| < 1$ , то запись числа называется нормализованной слева. Следующие примеры показывают, как можно представить любое число в формате с плавающей точкой:

а) в десятичной системе счисления:

$$372.95 = 0.37295 \times 10^3;$$

$$25 = 0.025 \times 10^3 = 0.25 \times 10^2;$$

$$0.0000015 = 0.15 \times 10^{-5} = 0.015 \times 10^{-4};$$

б) в двоичной системе счисления:

$$11010,1101 = 0,0110101101 \times 2^6 = 0,110101101 \times 2^5;$$

$$0,011011 = 0,11011 \times 2^{-1};$$

$$0,1 = 0,1 \times 2^0$$

(здесь порядок определяет, на сколько разрядов необходимо осуществить сдвиг относительно запятой).

Из этих примеров видно, что представление чисел в формате с плавающей точкой неоднозначно. В компьютере с целью минимизации погрешности при вычислениях и эффективного использования памяти применяется процедура нормализации справа.

Число называется *нормализованным справа*, если после запятой в мантиссе стоит не нуль. Например, числа  $0,0007610$  и  $0,000112$ , представленные соответственно в виде  $0,076 \times 10^{-2}$  и  $0,011 \times 2^{-2}$ , не являются нормализованными справа, а эти же числа, представленные соответственно в виде  $0,76 \times 10^{-3}$  и  $0,11 \times 2^{-3}$ , являются таковыми.

В дальнейшем под нормализацией числа будем понимать нормализацию справа.

Нормализованное число одинарной точности, представленное в формате с плавающей точкой, записывается в память следующим образом:

- знак числа – в бите 15 первого слова (0 – для положительных и 1 – для отрицательных чисел);
- порядок размещается в битах 7 – 14 первого слова, а мантисса занимает остальные 23 бита в двух словах.

Нормализованное число двойной точности записывается в четыре слова памяти и отличается от представления чисел с одинарной точностью только тем, что продолжение мантиссы размещается в следующих за первым словом трех последовательных словах памяти, а всего под мантиссу в этом случае отводится 55 бит.

Порядок числа, представленного в формате с плавающей точкой, изменяется в диапазоне от  $-128_{10}$  до  $+127_{10}$  и запоминается увеличенным на  $128_{10}$ . Такой способ представления порядка называется смещенным.

Следует иметь в виду, что, хотя для мантииссы отведено 23 разряда для чисел одинарной точности и 55 разрядов – для чисел двойной точности, в операциях участвует 24 и 56 разрядов соответственно, так как старший разряд мантииссы нормализованного числа не хранится, т.е. имеется так называемый скрытый разряд. Однако при аппаратном выполнении операций этот разряд автоматически восстанавливается и учитывается. Порядок числа также учитывает скрытый старший разряд мантииссы.

Также заметим, что нормализованная мантиисса в двоичной системе счисления всегда представляется десятичным числом  $m$ , лежащим в диапазоне:

$$0,5 < m < 1.$$

При выполнении арифметических операций над числами, представленными в формате с плавающей точкой, надо отдельно выполнять их для порядков и мантиисс:

- при алгебраическом сложении чисел надо сначала уравнивать порядки слагаемых;
- при умножении порядки надо складывать, а мантииссы – перемножать;
- при делении из порядка делимого вычитают порядок делителя, а над мантииссами совершают обычную операцию деления.

После выполнения операций, если это необходимо, проводят нормализацию результата, что влечет изменение порядков, так как каждый сдвиг на один разряд влево соответствует уменьшению порядка на единицу, а сдвиг вправо – увеличению его на единицу. Введение термина «плавающая точка» как раз и объясняется тем, что двоичный порядок, определяющий фактическое положение точки в изображении числа, корректируется после выполнения каждой арифметической операции, т.е. точка в изображении числа «плавает» (изменяется ее положение) по мере изменения данной величины. А в изображении чисел, представленных в формате с фиксированной точкой, она жестко зафиксирована в определенном месте.

Арифметические операции с числами, представленными в формате с плавающей точкой, намного сложнее таких же операций для чисел, представленных в формате с фиксированной точкой. Но зато плавающая точка позволяет производить операции масштабирования автоматически в самой машине и избавляет от накопления абсолютной погрешности при вычислениях (хотя не избавляет от накопления относительной погрешности).

### 2.3.5. ПРЕДСТАВЛЕНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Последовательностями нулей и единиц можно закодировать и графическую информацию. Различают три вида компьютерной графики: растровую, векторную и фрактальную. Рассмотрим наиболее часто используемую при разработке электронных (мультимедийных) и полиграфических изданий растровую графику. Основным элементом растрового изображения является точка, или пиксель.

Для кодирования любого изображения нужно разбить его на точки и цвет каждой точки закодировать. Например, черно-белую картинку можно закодировать, используя два бита: 11 – белый цвет, 10 – светло-серый, 01 – темно-серый и 00 – черный.

Для кодирования 256 различных цветов требуется 8 бит. Однако этого недостаточно для кодирования полноцветных изображений живой природы. Человеческий глаз может различать десятки миллионов цветовых оттенков. В современных компьютерах для кодирования цвета одной точки используется три байта.

Каждый цвет представляет собой комбинацию трех основных цветов: красного, зеленого и синего. Первый байт определяет интенсивность красной составляющей, второй – зеленой, третий – синей.

Белый цвет кодируется полными тремя байтами (255, 255, 255, или – в двоичной системе – 11111111, 11111111, 11111111). Черный цвет – отсутствие всех цветов (0, 0, 0). Красный цвет может быть темным (120, 0, 0) или ярко-красным (255, 0, 0). Такая система кодирования цветной графической информации называется системой **RGB** (Red, Green, Blue) и обеспечивает однозначное определение 16,5 млн ( $2^{24}$ ) различных цветов и оттенков.

Качество графического изображения зависит от количества точек (пикселей) на единице площади. Этот параметр называется разрешением и измеряется в точках на дюйм (*dpi*).

Расчет объема графической информации сводится к вычислению произведения количества точек на изображении на количество разрядов, необходимых для кодирования цвета одной точки. Например, для цветной картинки, составленной из 256 цветов в графическом режиме монитора  $640 \times 480$ , требуется объем видеопамати, равный:

$$8 \times 640 \times 480 = 2\,457\,600 \text{ бит} = 307\,200 \text{ байт} = 300 \text{ Кбайт.}$$

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Назовите основные способы измерения информации.
2. В чем состоит суть энтропийного подхода к измерению информации?
3. Что такое бит и байт?
4. Что такое машинное слово?
5. Что такое элемент и ячейка памяти?
6. Что такое основание системы счисления?
7. Какие числа называются базисными?
8. Какой смысл вкладывается в понятие записи и представления числа в системе счисления?
9. Что такое позиционная система счисления?
10. В чем состоит отличие позиционной системы счисления от непозиционной? Приведите примеры.
11. Назовите общее правило перевода чисел из любой системы счисления в десятичную систему.
12. Назовите правила перевода чисел из десятичной системы счисления в любую другую систему.
13. Сформулируйте основные свойства двоичной системы счисления в компьютере.
14. Какие операции с двоичными числами может выполнять процессор вычислительного устройства?
15. Какие существуют формы представления отрицательного числа в двоичной системе счисления?
16. Как представляются целые и действительные числа в компьютере? Приведите примеры.
17. Почему при выполнении арифметических операций возникают ситуации, приводящие к неверному результату?
18. Что такое дополнение до единицы и дополнение до двух?
19. Зачем нужны индикаторы переноса и переполнения?
20. Сформулируйте правило, определяющее выполнение операции сложения в разных числовых системах (со знаком и без знака).
21. Как представляются целые и вещественные числа в памяти компьютера?
22. Какой способ представления порядка числа с плавающей точкой называется смещенным?
23. Как представляются символные данные в памяти компьютера?
24. Что такое пиксель?
25. Какие данные хранятся в файлах, содержащих растровые изображения?

## 3. КОМПЬЮТЕР

### 3.1. ОСНОВНЫЕ УСТРОЙСТВА КОМПЬЮТЕРА

#### 3.1.1. ОБЩАЯ СХЕМА ФУНКЦИОНИРОВАНИЯ КОМПЬЮТЕРА

Создание и функционирование информационных систем в управлении экономикой тесно связано с развитием информационной технологии – главной составной части автоматизированной информационной системы.

**Автоматизированная информационная технология** (АИТ) – это системно организованная для решения задач управления совокупность методов и средств реализации операций сбора, регистрации, передачи, накопления, поиска, обработки и защиты информации на базе применения развитого программного обеспечения, используемых средств вычислительной техники и связи, а также способов, с помощью которых информация предлагается пользователю.

Все возрастающий спрос на информационные услуги привел к тому, что современная технология обработки информации ориентирована на применение самого широкого спектра технических средств и прежде всего электронных вычислительных машин и средств коммуникаций. На их основе создаются вычислительные системы и сети различных конфигураций с целью не только накопления, хранения, переработки информации, но и максимального приближения терминальных устройств к рабочему месту специалиста или принимающего решения руководителя.

Появление компьютеров и стремительное совершенствование их эксплуатационных возможностей создало реальные предпосылки для автоматизации управленческого труда, формирования рынка информационных продуктов и услуг. Развитие АИТ шло параллельно с появлением новых видов технических средств обработки и передачи информации, совершенствованием организационных форм использования компьютера, насыщением инфраструктуры экономики новым средством коммуникации.

Компьютеры появились, когда возникла острая необходимость в проведении трудоемких и точных расчетов. Уровень прогресса в таких областях науки и техники, как, например, атомная энергетика, аэрокосмические исследования, во многом зависел от возможности выполнения сложных расчетов, которые нельзя было осуществить на электромеханических счетных машинах. Требовался переход к вычислительным машинам, работающим с большой производительностью.

#### 3.1.2. ОСНОВНЫЕ БЛОКИ И УСТРОЙСТВА КОМПЬЮТЕРА

Чтобы понять назначение и взаимосвязь основных устройств компьютера, необходимо рассмотреть последовательность прохождения информации при ее обработке.

В основе организации вычислительного процесса на компьютере лежит принцип **программного управления**. Для решения задачи необходимо составить программу. **Программа** – это определенная последовательность команд (инструкций), которая обеспечивает выполнение задачи. Пользователь записывает программы на каком-либо алгоритмическом языке. Однако компьютер работает под управлением программы, переведенной с алгоритмического на машинный язык, который является собственным языком программирования. Машинным языком принято считать способ представления программ и исходных данных в доступном для компьютера виде, т.е. это формальный язык, предназначенный для описания алгоритма решения задачи, содержание и правила которого реализуются аппаратными средствами конкретной машины. Машинный язык строится на базе системы команд, характерной для данного компьютера.

Программа кодируется в цифровой форме, как и любая информация.

Назначение устройства ввода информации (УВв) – преобразование информации, представленной в символах входного алфавита, в символы внутреннего алфавита, в котором осуществляются обработка и хранение информации в компьютере.

Основной функцией *устройств хранения информации в компьютере* (памяти) является хранение программ и данных, необходимых для решения задачи, а также промежуточных и окончательных результатов вычисления. Пользователя интересуют прежде всего такие характеристики, как емкость памяти и ее быстродействие. Емкость памяти определяется максимально возможным количеством хранящейся информации. Быстродействие памяти характеризуется временем, которое необходимо для поиска (выборки), записи или считывания информации.

На практике емкость запоминающих устройств обычно измеряется в мегабайтах и гигабайтах.

Время обращения – интервал времени между началом и окончанием ввода (вывода) информации в память (из памяти). В отношении устройств памяти вместо термина «ввод-вывод» используют термин «запись-чтение». Таким образом, время обращения характеризует затраты времени на поиск места в памяти и запись (чтение) информации.

Различают оперативные и внешние запоминающие устройства (ЗУ). *Оперативные запоминающие устройства* (ОЗУ) характеризуются малым значением времени обращения при емкости от сотен мегабайт до десятков гигабайт. *Внешние запоминающие устройства* (ВЗУ) характеризуются в сравнении с ОЗУ большим временем обращения, но емкость их может быть практически неограниченной, так как их количество можно наращивать по мере необходимости.

С понятием «поиск места в памяти» связано понятие «адрес информации». Память компьютера можно представить состоящей из отдельных ячеек, в каждой из которых хранится определенное количество информации: один байт, два байта, четыре байта и т.д. Все ячейки в памяти пронумерованы. Номера ячеек называются адресами.

В современных компьютерах минимальные адресуемые элементы памяти обычно имеют емкость один байт. Если информация занимает в памяти несколько последовательных элементов, то она составляет поле памяти. Местонахождение информации в этом случае задается адресом первого байта и количеством занимаемых ею байт памяти.

Различают запоминающие устройства произвольного и последовательного доступа. Устройства произвольного доступа позволяют по заданному адресу одинаково быстро извлекать информацию из любого элемента памяти. Такими устройствами являются используемые в компьютере оперативные ЗУ. Примером запоминающих устройств последовательного доступа являются ЗУ на магнитных лентах (МЛ). Данные на МЛ записываются байт за байтом по всей длине. Чтобы считать информацию, необходимо перематывать ленту до тех пор, пока участок с записью требуемой информации не будет подведен к считывающему устройству. Время обращения при этом зависит от того, как «далеко» на ленте находится требуемая информация.

*Арифметико-логическое устройство* (АЛУ) выполняет арифметические и логические операции, предусмотренные системой команд компьютера, и характеризуется:

- набором выполняемых элементарных операций (сложением, вычитанием, умножением, делением, сравнением, логическими операциями);
- временем выполнения элементарных операций;
- средним быстродействием, т.е. количеством арифметических и логических операций, выполняемых в единицу времени (секунду).

*Устройство управления* (УУ) предназначено для автоматического управления ходом вычислительного процесса и обеспечивает необходимое взаимодействие всех устройств машины по автоматическому выполнению заданного алгоритма решения задачи. Функции УУ, как наиболее важные для понимания автоматической работы ком-

пьютера, описываются далее при рассмотрении реализации в компьютере принципа программного управления.

**Устройства вывода** предназначены для вывода информации из машины и преобразования результатов решения задач из символов внутреннего алфавита в выходную информацию, представленную в символах выходного алфавита. В качестве устройств вывода применяют печатающие устройства, графопостроители, дисплеи, различные звукогенерирующие устройства. В качестве устройства вывода компьютера можно рассматривать и съемные накопители информации.

### 3.1.3. АРХИТЕКТУРА КОМПЬЮТЕРА

Большинство современных компьютеров строится на базе принципов, сформулированных американским ученым, одним из «отцов» кибернетики Джоном фон Нейманом (1903 – 1957 гг.). Впервые эти принципы были опубликованы фон Нейманом в 1945 г. в его предложениях по машине EDVAC. Этот компьютер была одной из первых машин с хранимой программой, т.е. с программой, находящейся в памяти машины, а не считываемой с перфокарты или другого подобного устройства.

В целом **принципы фон Неймана** сводятся к следующему:

- 1) основными блоками фоннеймановской машины являются блок управления, арифметико-логическое устройство, память и устройство ввода-вывода;
- 2) информация кодируется в двоичной форме и разделяется на единицы, называемые словами;
- 3) алгоритм представляется в форме последовательности управляющих слов, которые определяют смысл операции. Эти управляющие слова называются командами. Совокупность команд, представляющая алгоритм, называется программой;
- 4) программы и данные хранятся в одной и той же памяти. Разнотипные слова различаются по способу использования, но не по способу кодирования;
- 5) устройство управления и АЛУ обычно объединяются в одно устройство, называемое центральным процессором (ЦП). Центральный процессор определяет действия, подлежащие выполнению, путем считывания команд из оперативной памяти. Обработка информации, предписанная алгоритмом, сводится к последовательному выполнению команд в порядке, однозначно определяемом программой.

Принципы фон Неймана практически можно реализовать множеством различных способов. Рассмотрим один из них: компьютер с шинной, или магистрально-модульной, организацией. Введем несколько определений.

**Архитектура компьютера** – абстрактное определение машины в терминах основных функциональных модулей, языка, структур данных. Архитектура не определяет особенности реализации аппаратной части компьютера, время выполнения команд, степень параллелизма, ширину шины и другие аналогичные характеристики. Архитектура отображает структуру компьютера, которая является видимой для пользователя: систему команд, режимы адресации, форматы данных, набор программно-доступных регистров. Одним словом, термин «архитектура» используется для описания возможностей, представляемых компьютером. Весьма часто употребляется термин «конфигурация» компьютера, под которым понимается компоновка вычислительного устройства с четким определением характера, количества взаимосвязей и основных характеристик его функциональных элементов. Термин «организация» компьютера определяет, как реализованы возможности компьютера.

**Команда** – совокупность сведений, необходимых процессору для осуществления определенного действия при выполнении программы. Команда состоит из кода операции, содержащего указание на операцию, которую необходимо выполнить, и нескольких адресных полей, содержащих указание на места расположения операндов команды.

Способ вычисления адреса по информации, содержащейся в адресном поле команды, называется режимом адресации. Множество команд, реализованных в данном компьютере, образует систему команд.

### 3.1.4. МАГИСТРАЛЬНО-МОДУЛЬНЫЙ ПРИНЦИП ПОСТРОЕНИЯ КОМПЬЮТЕРА

Магистрально-модульная (шинная) организация является простейшей формой организации компьютера. В соответствии с принципами фон Неймана компьютер имеет в своем составе следующие функциональные блоки (рис. 3.1).

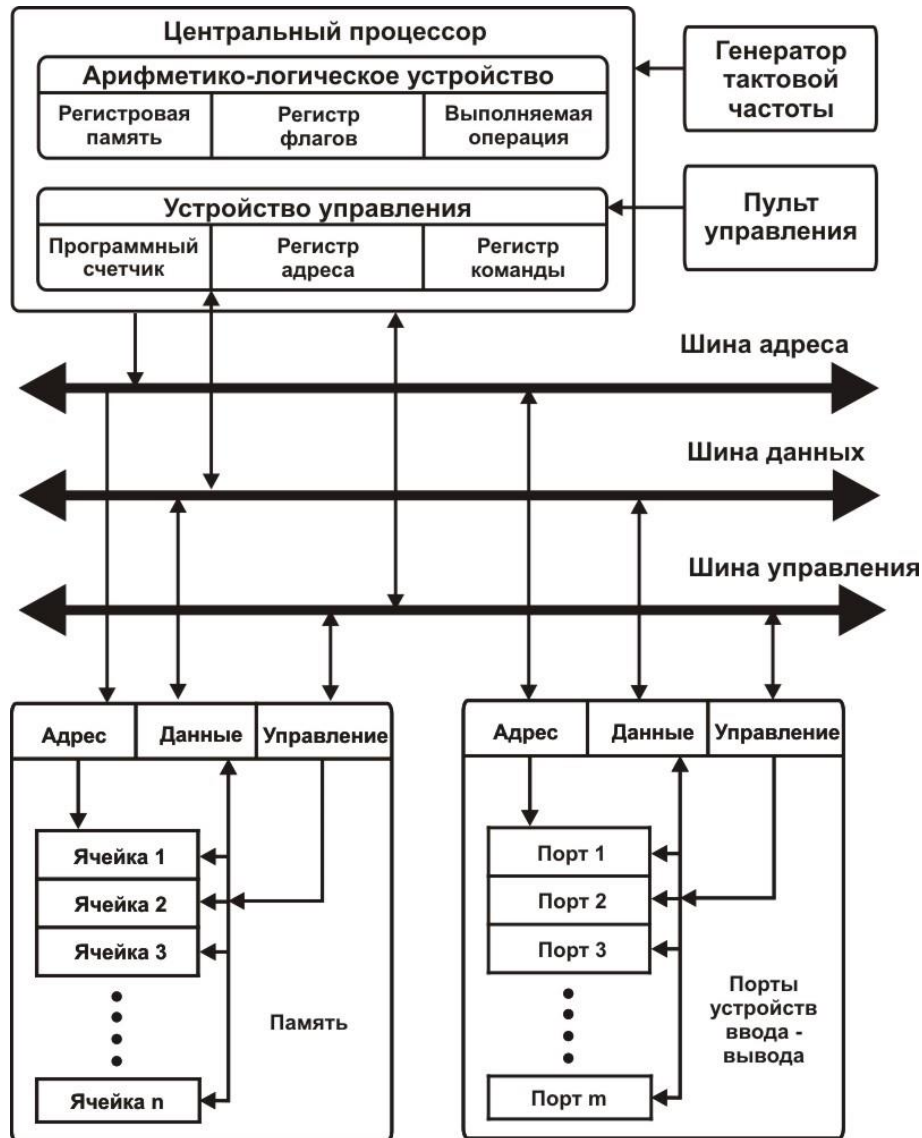


Рис. 3.1. Схема ЭВМ с шинной организацией

Магистраль (наиболее часто употребляется термин «шина») – это совокупность электрических линий для обмена данными между устройствами компьютера. Тип шины определяет и сигналы, которые передаются по этим линиям. В персональном компьютере типы шин определяются материнской платой. Основные характеристики шин – *разрядность* передаваемых данных и *скорость передачи* данных в мегабайтах в секунду (Мбайт/с). Наибольший интерес вызывают два типа шин: системный и локальный. Системная шина предназначена для обеспечения передачи данных между периферийными устройствами и центральным процессором, а также оперативной памятью

(ОП). Локальная шина – это шина, непосредственно подключенная к контактам микропроцессора, т. е. шина процессора.

**Модуль** – функциональная часть технического обеспечения вычислительной системы, выполненная в одном блоке и имеющая узлы сопряжения с другими модулями. Модули применяются для удобной эксплуатации вычислительных сетей (ВС), их можно быстро заменить при выходе из строя и при модернизации.

**Центральный процессор** – функциональная часть компьютера, выполняющая основные операции по обработке данных и управлению работой других блоков. Это наиболее сложный компонент компьютера.

**Устройство управления (УУ)** – часть центрального процессора. Оно вырабатывает распределенную во времени и пространстве последовательность внутренних и внешних управляющих сигналов, обеспечивающих выборку и выполнение команд. На этапе цикла выборки команды УУ интерпретирует команду, выбранную из программной памяти. На этапе выполнения команды в соответствии с типом реализуемой операции УУ формирует требуемый набор команд низкого уровня для АЛУ и других устройств. Эти команды задают последовательность простейших низкоуровневых операций, таких, как пересылка данных, сдвиг данных, установка и анализ признаков, запоминание результатов и др. Такие элементарные низкоуровневые операции называются микрооперациями, а команды, формируемые устройством управления, – микрокомандами.

**Память** – устройство, предназначенное для запоминания, хранения и выборки программ и данных. Память состоит из конечного числа ячеек, каждая из которых имеет свой уникальный номер, или адрес. Доступ к ячейке осуществляется указанием ее адреса. Память способна выполнять два вида операций над данными – чтение с сохранением содержимого и запись нового значения со стиранием предыдущего. Как уже говорилось, каждая ячейка памяти может использоваться для хранения либо порции данных, либо команды. В большинстве современных компьютеров минимально адресуемым элементом памяти является байт – поле из восьми бит. Совокупность бит, которые АЛУ может одновременно поместить в регистр или обработать, называется обычно машинным словом.

**Оперативная память (ОП)** – функциональный блок, хранящий информацию для УУ (команды) и АЛУ (данные).

**Периферийные устройства (ПУ)**. В их число входят устройства внешней памяти, предназначенные для долговременного хранения данных большого объема и программ, и коммуникационные устройства, предназначенные для связи компьютера с внешним миром (с пользователем, другими компьютерами и т.д.). Обмен данными с внешним устройством осуществляется через порты ввода-вывода. «Порт» (от англ. *port* – ворота, дверь, отверстие) – абстрактное понятие. По аналогии с ячейками памяти порты можно рассматривать как ячейки, через которые можно записать информацию в ПУ или, наоборот, прочесть ее из него. Так же, как и ячейки памяти, порты имеют уникальные номера – адреса портов ввода-вывода.

Объединение функциональных блоков в компьютере осуществляется посредством следующей **системы шин**:

- шины данных, по которой осуществляется обмен информацией между блоками компьютера;
- шины адреса, используемой для передачи адресов (номеров ячеек памяти или портов ввода-вывода, к которым производится обращение);
- шины управления, которая служит для передачи управляющих сигналов.

Совокупность этих трех шин называется **системной шиной**, системной магистралью или системным интерфейсом. Состав и назначение шин, правила их использования, виды передаваемых по шине сигналов и другие характеристики шины могут су-

шественно различаться у разных видов компьютеров. Однако есть принципиально общие закономерности организации шин.

Шина состоит из отдельных проводников (линий). Сигналы по линиям шины могут передаваться либо импульсами (наличие импульса соответствует логической единице, а отсутствие импульса – нулю), либо уровнем напряжения (например, высокий уровень – логическая единица, низкий – нуль). Количество линий (проводников), входящих в состав шины, называется шириной шины. Ширина шины адреса определяет

размер адресного пространства компьютера. Если, например, количество линий адреса, используемых для адресации памяти, равно 20, то общее количество адресуемых ячеек памяти составит  $2^{20}$ , т. е. немногим более 1 млн (точнее – 1 048 576) ячеек.

Обычно на шине в любой момент можно выделить два активных устройства. Одно из них называется задатчиком и инициирует операцию обмена данными (формирует адреса и управляющие сигналы), другое – исполнителем и выполняет операцию (формирует адреса и управляющие сигналы и принимает или передает данные). В большинстве случаев задатчиком является ЦП. Память всегда выступает только в качестве исполнителя.

Функционирование компьютера с шинной структурой можно описать следующим обобщенным алгоритмом (рис. 3.2):

1) после включения компьютера или операции сброса в регистры центрального процессора заносятся некоторые начальные значения (инициализация). Обычно в процессе инициализации в память компьютера помещается программа, называемая первичным за-



Рис. 3.2. Обобщенный алгоритм работы ЭВМ

грузчиком. Основное назначение первичного загрузчика – загрузить в память с устройства внешней памяти операционную систему. Эта программа может быть размещена в энергонезависимом устройстве памяти или автоматически считываться с некоторого устройства внешней памяти. Мы не будем здесь подробно останавливаться на механизмах загрузки операционной системы, тем более что они могут существенно различаться для разных типов компьютеров. Пока будем полагать, что в памяти некоторым образом оказалась первая из подлежащих выполнению программ. Программному счетчику присваивается начальное значение, равное адресу первой команды программы;

2) центральный процессор производит операцию считывания команды из памяти. В качестве адреса ячейки памяти используется содержимое программного счетчика;

3) содержимое считанной ячейки памяти интерпретируется процессором как команда и помещается в регистр команды. Устройство управления приступает к интерпретации прочитанной команды. По полю команды операции из первого слова команды устройство управления определяет ее длину и, если это необходимо, организует дополнительные операции считывания, пока вся команда полностью не будет прочитана процессором. Вычисленная длина команды прибавляется к исходному содержимому программного счетчика, и когда команда полностью прочитана, программный счетчик будет хранить адрес следующей команды;

4) по адресным полям команды устройство управления определяет, имеет ли команда операнды в памяти. Если имеет, то на основе указанных в адресных полях режимов адресации вычисляются адреса операндов и производятся операции чтения памяти для считывания операндов;

5) устройство управления и арифметико-логическое устройство выполняют операцию, указанную в поле кода операции команды. Во флаговом регистре процессора запоминаются признаки результата операции (равно нулю или нет, знак результата, наличие переполнения и т.д.);

6) если это необходимо, устройство управления выполняет операцию записи, для того чтобы поместить результат выполнения команды в память;

7) если последняя команда не была командой «*Остановить процессор*», то описанная последовательность действий повторяется, начиная с шага 1.

Описанная последовательность действий центрального процессора с шага 1 до шага 6 называется циклом процессора.

### 3.1.5. ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕР

С 1975 г. в США было начато серийное производство персонального компьютера (ПК). Это событие сегодня в мире часто называют второй информационной революцией (первой информационной революцией считается появление печатного станка и книгопечатания – 1445 г.). Персональный компьютер появился на базе мини- и микро-ЭВМ, он обеспечивает персональные вычисления, т.е. режим работы специалиста в данной предметной области, непосредственно на рабочем месте. За дисплей компьютера смог сесть пользователь – непрофессионал в программировании. С 1981г. стали выпускаться ПК, имеющие блочно-модульную конструкцию. Эти простые в эксплуатации и сравнительно дешевые машины предназначались для потребителей, не обладающих знаниями в области вычислительной техники и программирования. Широкое распространение мини-ЭВМ в начале 1970-х гг. определялось необходимостью приблизить компьютер к пользователю. Мини-ЭВМ устанавливались непосредственно на предприятиях и в организациях, где использование больших компьютеров было экономически невыгодным.

Таким образом, ПК – это компьютер, предназначенный для индивидуального использования. В настоящее время это мощный универсальный компьютер. Он успешно работает как дома, так и на рабочих местах в офисах, легко подключается к различным вычислительным сетям.

Основные критерии отнесения компьютера к классу ПК – малые размеры, отсутствие необходимости в обслуживании, низкая цена, функциональная универсальность и простота модернизации.

Важную роль в развитии ПК сыграло появление компьютера IBM PC, произведенного корпорацией IBM (США) на базе МП Intel-8086 в 1981 г. Этот персональный компьютер занял ведущее место на рынке ПК. Его основное преимущество – так называемая *открытая архитектура*, благодаря которой пользователи могут расширять возможности приобретенного ПК, добавляя различные периферийные устройства и модернизируя компьютер. В дальнейшем другие фирмы начали создавать свои ПК, но компьютер IBM PC стал как бы стандартом класса ПК. В наши дни около 85 % всех продаваемых ПК базируется на архитектуре IBM PC.

### 3.2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА

Функционирование любого компьютера требует различных видов обеспечения: математического, алгоритмического, информационного, программного, технологического, лингвистического, организационного, эргономического, правового и др.

Важнейшее значение имеет программное обеспечение (ПО) компьютера. Программное обеспечение (англ. *software*) – это совокупность программных средств для обеспечения нормальной работы компьютера. Программное обеспечение любого компьютера в основном определяет его интеллектуальные возможности, профессиональную направленность, широту и полноту осуществления устройств (блоков).

Программное обеспечение подразделяют на *системное* (общее) и *прикладное* (специальное ПО). Основные элементы системного ПО обычно поставляются вместе с компьютером. К ним относятся операционные системы (ОС) и программы, расширяющие возможности ОС. Главное предназначение системного ПО – управление работой процессора, организация интерфейса между пользователем и компьютером, организация доступа к периферийным устройствам и сети, управление файлами, запуск прикладных программ и управление процессом их выполнения, трансляция и выполнение программ, подготовленных на различных алгоритмических языках.

Прикладное ПО, как правило, состоит из уникальных программ и функциональных пакетов прикладных программ. Именно от функционального ПО зависят вид, содержание и конкретная специализация пользователя. Учитывая, что прикладное ПО в конечном счете определяет область применения компьютера и состав решаемых пользователем задач, оно должно создаваться на основе инструментальных программных средств диалоговых систем, ориентированных на решение конкретного класса задач со схожими функционально-техническими особенностями обработки информации.

Прикладное ПО должно обладать свойствами адаптивности (приспосабливаемости), гибкости, модифицируемости и настраиваемости на конкретное применение в соответствии с требованиями пользователя.

**Операционная система** – это комплекс программ, обеспечивающий управление компьютером как единым целым (тогда как на самом деле компьютер состоит из многих частей), осуществляющий взаимодействие компьютера с окружающей средой, а именно: с человеком, прикладными программами, другими системами. Операционная система – главная часть системного ПО.

Программные средства, расширяющие возможности ОС, – это ПС, расширяющие возможности таких операционных систем, которые обеспечивают функционирование компьютера различных конфигураций. К ним относятся пакеты, обеспечивающие работу многомашинных комплексов типовых конфигураций, диалоговые системы, системы для работы в реальном масштабе времени, удаленную пакетную обработку.

Программные средства защиты информации – это развитые программные комплексы, обеспечивающие защиту информации от всех видов постороннего вмешательства.

Наиболее динамично развивающаяся часть прикладного ПО – **пакеты прикладных программ** (ППП). Круг решаемых с их помощью задач постоянно расширяется. Во многом внедрение компьютеров практически во все сферы деятельности стало возможным благодаря появлению новых и совершенствованию существующих ППП.

Достижения в области микроэлектроники, появление более мощных по своим функциональным возможностям компьютеров также стимулируют создание новых ППП. В свою очередь, необходимость улучшения характеристик ППП при решении конкретных задач пользователя стимулирует совершенствование архитектуры и элементной базы компьютеров и периферийных устройств.

Каждая группа ППП имеет свои проблемы организации, трудности разработки и создания. Каждый пакет реализуется в соответствии с предъявляемыми к нему требованиями и возможностями конкретного языка программирования.

Пакеты прикладных программ общего назначения включают набор программ для графических дисплеев, графопостроителей, систем программирования, а также для

научно-технических расчетов, математического программирования, обработки матриц, различного вида моделирования, решения задач теории массового обслуживания и т.д.

Пакеты прикладных программ, ориентированные на работу в АСУ, включают набор программ для общецелевых систем обработки банков данных, информационных систем общего назначения, систем обработки документов.

Существующие ППП охватывают почти все сферы человеческой деятельности, связанные с обработкой информации. Развитие и совершенствование ППП – поступательный процесс, поэтому следует ожидать появления новых пакетов, возможности которых превзойдут достижения существующих.

По функциональному признаку ППП подразделяют на проблемно-ориентированные и интегрированные.

**Проблемно-ориентированные ППП** – наиболее развитая в плане реализуемых функций и многочисленная по количеству созданных пакетов часть ПО. Она включает следующие проблемно-ориентированные программные продукты: текстовые процессоры, издательские системы, графические редакторы, демонстрационную графику, системы мультимедиа, организаторы работ, электронные таблицы (табличные процессоры), системы управления базами данных, программы распознавания символов, финансовые и аналитико-статистические программы.

**Интегрированные ППП** – немногочисленная по количеству наименований продуктов, но довольно мощная в вычислительном плане и активно развивающаяся часть ПО. Интегрированные ППП подразделяют на две группы: традиционные, или полностью связанные, и объектно-связанные.

Традиционные, или полностью связанные, интегрированные комплексы представляют собой многофункциональный автономный пакет, в котором в одно целое соединены функции и возможности различных специализированных (проблемно-ориентированных) пакетов, родственных в смысле технологии обработки данных на отдельном рабочем месте. Типичным представителем таких программ является пакет Microsoft Word.

В настоящее время активно реализуется другой подход к интеграции программных средств: объединение специализированных пакетов в рамках единой ресурсной базы, обеспечение взаимодействия приложений (программ пакета) на уровне объектов и единого упрощенного центра – переключения между приложениями. Интеграция в этом случае носит объектно-связанный характер.

Типичные и наиболее мощные пакеты данного типа – Microsoft Office, в различных редакциях которого присутствуют: текстовый редактор, система управления базой данных (СУБД), табличный процессор, программы демонстрационной графики. Целесообразность создания таких пакетов, очевидно, связана с желанием получить дополнительный эффект от интеграции по отношению к простой сумме составляющих его компонентов. Этот эффект должен достигаться за счет согласованного взаимодействия компонентов в процессе работы пользователя.

### **3.3. ФАЙЛОВАЯ СИСТЕМА. РАБОТА С НОСИТЕЛЯМИ ИНФОРМАЦИИ**

#### **3.3.1. ФАЙЛОВАЯ СИСТЕМА**

Для обеспечения удобства работы с записанными на диск сведениями их размещают в файлах. **Файл** – это логически связанная совокупность данных, для которой во внешней памяти отводится поименованная область (данные – это любая информация, включающая программу и исходные данные для их выполнения, результаты выполнения программ, тексты, иллюстрации и т. п.).

Обычно в отдельном файле хранят данные, относящиеся к одному типу. Тип данных определяет тип файла. Файл хранится в виде последовательности произвольного

числа байт, обладающей уникальным именем. Файл может содержать любое число байт или быть пустым (0 байт). Создать файл – значит присвоить ему имя. Уникальность имени файла гарантирует однозначность доступа к данным.

Имя файла задается по следующим правилам: составное (полное) имя файла представляет собой совокупность собственно имени файла и расширения имени файла; имя от расширения отделяется точкой.

Расширение имени файла передает операционной системе информацию о том, к какому типу относятся данные, содержащиеся в файле, и о формате, в котором они записаны.

Файл характеризуют следующие параметры (свойства):

- полное имя;
- объем в байтах;
- дата создания;
- время создания;
- атрибуты файла, которые определяют степень доступа к нему: R (Readonly) – только для чтения, H (Hidden) – скрытый, S (System) – системный файл, A (Archive) – архивированный файл.

**Файловая система** – это функциональная часть операционной системы, обеспечивающая хранение данных на дисках и доступ к ним. Файловая система определяет способы организации и средства обслуживания файловой структуры.

К операциям, выполняемым операционной системой по обслуживанию файловой структуры, относятся:

- создание файлов и присвоение им имен;
- создание каталогов и присвоение им имен;
- переименование файлов и каталогов;
- копирование и перемещение файлов и каталогов;
- удаление файлов и каталогов;
- навигация по файловой структуре с целью доступа к заданному файлу, каталогу;
- управление атрибутами файлов.

Для обеспечения удобного доступа к файлам файловая система позволяет объединять их в каталоги. Каталогом называется специальный файл, в котором регистрируются другие файлы и каталоги. Если файл зарегистрирован в каталоге, это означает, что в последнем содержатся вся характеризующая файл информация и сведения о том, в каком месте диска файл расположен. Сам же файл хранится как последовательность байт без каких-либо дополнительных справочных сведений.

Правила именования каталогов совпадают с правилами именования файлов, однако расширения практически не используются. Каталоги низких уровней вкладываются в каталоги более высоких уровней и являются для них вложенными. Верхним уровнем вложенности иерархической структуры является корневой каталог.

На каждом диске всегда имеется единственный корневой каталог (он именуется символом «\»), в который могут входить другие каталоги и файлы. Корневой каталог создается при форматировании (разметке) диска, хранится во вполне определенной области дисковой памяти, имеет ограниченный размер и не может быть удален никакими средствами. Пользователь не имеет возможности что-либо сделать с корневым каталогом, за исключением помещения в него файлов и других каталогов, а также удаления их из него.

С понятием файла и каталога в ОС связано понятие логического диска. Логический диск создается и управляется специальной программой, имеет уникальное имя в виде одной латинской буквы (например: C, D, E, F и т. д.), может быть реализован на

жестком и гибком дисках, на CD-ROM, в оперативной памяти (электронный диск). На одном физическом диске может быть создано несколько логических дисков.

Для обеспечения доступа к существующему файлу или для определения места размещения файла в файловой структуре в общем случае требуется задать:

- имя привода, на котором установлен диск, содержащий искомый файл или предназначенный для размещения нового файла;
- путь к файлу по файловой структуре этого диска;
- составное имя файла (имя файла и расширение имени файла).

Данные сведения указываются в спецификации файла, которая имеет следующий синтаксис (представление, форму, структуру):

[имя носителя]: [\маршрут \] имя\_файла.[расширение].

Маршрут (путь) – это цепочка соподчиненных каталогов, которую надо пройти по иерархической структуре к каталогу, где зарегистрирован искомый файл. При задании пути имена каталогов записываются в порядке следования и отделяются друг от друга символом «\». Здесь необязательные элементы заключены в квадратные скобки, так как операционная система хранит информацию о текущем диске в текущем каталоге.

**Операционная система имеет два режима работы:** пакетный и диалоговый. Пакетный режим состоит в том, что ОС автоматически исполняет заданную последовательность команд. В отличие от пакетного, при диалоговом режиме операционная система находится в состоянии ожидания команды пользователя; получив ее, приступает к исполнению, а исполнив – возвращает отклик и ждет очередной команды.

### 3.3.2. ГРАФИЧЕСКИЕ ПОЛЬЗОВАТЕЛЬСКИЕ ИНТЕРФЕЙСЫ

По типу пользовательского интерфейса различают символьные (линейные) и графические операционные системы.

Линейные ОС реализуют интерфейс командной строки. Основным устройством управления в данном случае является клавиатура. Команда набирается на клавиатуре и отображается на экране дисплея. Окончанием ввода команды служит нажатие клавиши «Enter». Для работы с операционными системами, имеющими линейный интерфейс, необходимо овладеть командным языком данной среды, т.е. совокупностью команд, структура которых определяется синтаксисом этого языка.

Графические ОС реализуют интерфейс, основанный на взаимодействии активных и пассивных графических экранных элементов управления. Устройствами управления в данном случае являются клавиатура и мышь. Активным элементом управления является указатель мыши – графический объект, перемещение которого на экране синхронизировано с перемещением мыши.

Пассивные элементы управления – это графические элементы управления приложений (экранные кнопки, значки, переключатели, флажки, раскрывающиеся списки, строки меню и т. д.).

Операционные системы семейства Windows (от англ. *windows* – окно) являются графическими операционными системами компьютеров платформы IBM PC.

Основным понятием операционной системы Windows является объект, его свойства и действия, которые он может выполнить в зависимости от запроса.

### 3.4. КОМПЬЮТЕРНЫЕ ВИРУСЫ И АНТИВИРУСНЫЕ ПРОГРАММЫ

#### 3.4.1. КОМПЬЮТЕРНЫЕ ВИРУСЫ

Большую опасность для компьютера представляют компьютерные вирусы. *Компьютерный вирус* – это специально написанная небольшая по размерам программа, которая может приписывать себя к другим программам (т. е. заражать их), а также выполнять различные нежелательные действия. Программа, внутри которой находится компьютерный вирус, называется зараженной. Когда такая программа начинает работу, то сначала управление получает вирус, который находит и заражает другие программы, а также выполняет ряд вредных действий, в частности засоряет активную память, портит файлы и т.д.

Для маскировки вируса его действия по заражению других программ и нанесению вреда могут выполняться не всегда, а при каких-либо условиях. После того как вирус выполнит нужные ему действия, он передает управление той программе, в которой он находится, и она работает как обычно, т. е. внешне работа зараженной программы какое-то время не отличается от работы незараженной.

Вирус может действовать быстро и без выдачи сообщений, поэтому пользователь часто не замечает, что компьютер работает несколько странно. Многие вирусы устроены так, что при запуске зараженной программы они остаются постоянно (точнее, до перезагрузки ОС) в памяти компьютера и время от времени заражают другие программы. Кроме того, зараженные программы с данного компьютера могут быть перенесены с помощью дискет или по сети на другие компьютеры.

Среди всех вирусов можно выделить следующие группы:

- загрузочные (от англ. *boot*) вирусы – заражают программу начальной загрузки компьютера, хранящуюся в загрузочном секторе диска и запускающуюся при загрузке компьютера;
- файловые вирусы – в простейшем случае заражают исполняемые файлы, но могут распространяться и через файлы документов (системы Word for Windows) и даже вообще не модифицировать файлы, а лишь иметь к ним какое-либо отношение;
- загрузочно-файловые вирусы – имеют признаки как загрузочных, так и файловых вирусов;
- драйверные вирусы – заражают драйверы устройств компьютера или запускают себя путем включения в файл конфигурации дополнительных строк;
- сетевые вирусы – распространяются в сетях, объединяющих сотни и тысячи компьютеров.

Создание компьютерных вирусов с юридической точки зрения можно квалифицировать как преступление.

#### 3.4.2. АНТИВИРУСНЫЕ ПРОГРАММЫ И ЗАЩИТА ИНФОРМАЦИИ

*Антивирусная программа* (от англ. *antivirus program*) – это программа, которая производит поиск и уничтожение вирусов. Антивирусные программы (или программные комплексы) также помогают восстановить испорченные файлы.

Известные в настоящее время антивирусные программы можно подразделить на несколько типов:

- «детекторы». Их назначение – обнаружить вирусы. Программы – «детекторы» вирусов могут сравнивать загрузочные сектора дисков с известными загрузочными секторами, формируемыми операционными системами различных версий, и таким образом обнаруживать загрузочные вирусы или выполнять сканирование файлов на магнитных дисках с целью обнаружения сигнатур известных вирусов. Такие программы в настоящее время в чистом виде редки;

---

- **«фаги»**, или программы-«доктора». «Фаг» (от греч. *phagos* – пожиратель) – это программа, которая способна не только обнаружить вирус, но и уничтожить его, т. е. она удалит его код из зараженных программ и восстановит их работоспособность;

- **«ревизоры»**. Программы-«ревизоры» относятся к самым надежным средствам защиты от вирусов и должны входить в арсенал каждого пользователя. «Ревизоры» запоминают исходное состояние программ, каталогов и системных областей диска тогда, когда компьютер не заражен вирусом, а затем периодически сравнивают текущее состояние с исходным;

- **«сторож»**. «Сторож» – резидентная программа, постоянно находящаяся в памяти компьютера, контролирующая операции, связанные с попыткой коррекции файлов с расширением .COM и .EXE, с изменением атрибутов файлов, их записи в загрузочные сектора диска, прямой записи на диск по абсолютному адресу и т. д. «Сторож» предупреждает пользователя об этих операциях, но не лечит зараженные программы;

- **«вакцины»**. «Вакцины», или «иммунизаторы», – это резидентные программы, предотвращающие заражение файлов. «Вакцины» применяются, если отсутствуют программы-«доктора», «лечащие» этот вирус. «Вакцинация» возможна только для известных вирусов. В настоящее время программы-«вакцины» имеют ограниченное применение;

- **Dr.Web**. Эта программа представляет собой классический «полифаг». Она производит сканирование файлов и системных областей дисков компьютера на наличие в них компьютерных вирусов и при нахождении последних производит их лечение. Dr.Web реализует эвристический метод поиска, который заключается в обнаружении фрагментов программ, типичных для компьютерных вирусов. Эвристический анализатор используется «полифагами» для обнаружения вирусов, не входящих в базу данных «полифага». Эффективность эвристического анализатора определяется двумя параметрами: процентом обнаруженных вирусов и процентом ложных срабатываний (подозрение на вирусы в файлах, в которых их нет). Dr.Web может находить и обезвреживать полиморфные вирусы (не имеющие определенной сигнатуры), проверять файлы, находящиеся в архивах.

Практика функционирования систем обработки информации и вычислительных систем доказала, что существует много возможных направлений утечки информации и путей несанкционированного доступа в системах и сетях. В их числе:

- чтение остаточной информации в памяти системы после выполнения санкционированных запросов;
- копирование носителей и файлов информации с преодолением мер защиты;
- маскировка под зарегистрированного пользователя;
- использование программных «ловушек»;
- использование недостатков операционной системы;
- незаконное подключение к аппаратуре и линиям связи;
- злоумышленный вывод из строя механизмов защиты;
- внедрение и использование компьютерных вирусов.

Обеспечение безопасности информации в вычислительных сетях и в автономно работающих компьютерах достигается комплексом организационных и организационно-технических мер, технических и программных средств.

К организационным мерам защиты информации относятся:

- ограничение доступа в помещения, в которых происходят подготовка и обработка информации;
- обеспечение доступа к обработке и передаче конфиденциальной информации только проверенных должностных лиц;
- хранение магнитных носителей и регистрационных журналов в закрытых для доступа посторонних лиц сейфах;

- исключение просмотра посторонними лицами содержания обрабатываемых материалов через дисплей, принтер и т. д.;
- использование криптографических кодов при передаче по каналам связи ценной информации;
- уничтожение бумаги и иных материалов, содержащих фрагменты ценной информации.

К организационно-техническим мерам защиты информации относятся: подключение оборудования, обрабатывающего ценную информацию, к питанию от независимого источника или через специальные фильтры;

- установка на дверях помещений кодовых замков;
- уничтожение информации, хранящейся в ОЗУ и на «винчестере», при списании или отправке компьютера в ремонт;
- установка клавиатуры и принтеров на мягкие прокладки с целью снизить возможность снятия информации акустическим способом;
- ограничение электромагнитного излучения путем экранирования помещений, где происходит обработка информации, листами из металла или специальной пластмассы.

Технические средства защиты информации – это системы охраны территорий и помещений с помощью экранирования машинных залов и контрольно-пропускные системы. Защита информации с помощью технических средств предусматривает:

- контроль доступа к различным уровням памяти компьютера;
- блокировку данных и ввод ключей;
- выделение контрольных бит для записей с целью идентификации и др.

Архитектура программных средств защиты информации включает:

- контроль безопасности, в том числе контроль регистрации вхождения в систему, фиксацию в системном журнале,
- контроль действий пользователя;
- реакцию, в том числе звуковую, на нарушение системы защиты контроля доступа к ресурсам сети;
- контроль мандатов доступа;
- формальный контроль защищенности операционных систем (базовой операционной и сетевой);
- контроль алгоритмов защиты;
- проверку и подтверждение правильности функционирования технического и программного обеспечения.

К механизмам обеспечения безопасности относятся:

- идентификация пользователей;
- шифрование данных;
- электронная подпись;
- управление маршрутизацией и др.

Идентификация пользователей позволяет устанавливать конкретного пользователя, работающего за терминалом и принимающего или отправляющего сообщения.

Право доступа к определенным вычислительным и информационным ресурсам, программам и наборам данных, а так же к вычислительным сетям в целом предоставляется ограниченному контингенту лиц. Система должна распознавать пользователей, работающих за терминалами. Идентификация пользователей чаще всего производится с помощью паролей.

**Пароль** – совокупность символов, известных подключенному к сети абоненту, – вводится в начале сеанса взаимодействия с сетью, а иногда и в конце сеанса (в особо ответственных случаях пароль выхода из сети может отличаться от входного).

Система может предусматривать ввод пароля для подтверждения правомочия пользователя через определенные интервалы времени. Вычислительная система определяет подлинность пароля и, тем самым, пользователя.

Для защиты средств идентификации пользователей от неправомерного применения пароли передаются и сравниваются в зашифрованном виде, таблицы паролей также хранятся в зашифрованном виде, что исключает возможность их прочтения без знания ключа.

Для идентификации пользователей могут применяться и карточки с магнитным покрытием, на которых записывается персональный идентификатор пользователя, или карточки со встроенным чипом. Чтобы уменьшить риск злоупотреблений, при вводе карточек также предусматривается какой-либо способ идентификации пользователя, например короткий пароль.

Наиболее надежным, хотя и наиболее сложным является способ идентификации пользователя на основе анализа его индивидуальных параметров: отпечатков пальцев, рисунков линий ладони, радужной оболочки глаз и др.

**Шифрование данных** – это обеспечение секретности методами криптографии, т. е. методами преобразования данных из общепринятой формы в кодированную (шифрование) и обратного преобразования (дешифрования) на основе правил, известных только взаимодействующим абонентам сети. Криптография применяется для защиты передаваемых данных, а также информации, хранимой в базах данных, на магнитных и оптических дисках и т. д.

К криптографическим средствам предъявляется требование сохранения секретности, даже когда известна сущность алгоритмов шифрования-дешифрования. Секретность обеспечивается введением в алгоритмы специальных ключей (кодов).

Зашифрованный текст превращается в исходный только в том случае, когда в процессе шифрования и дешифрования используется один и тот же ключ. Область значений ключа выбирается столь большой, что практически невозможно его определить путем простого перебора.

### **3.5. ИСТОРИЯ И ПЕРСПЕКТИВЫ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ**

#### **3.5.1. ИСТОРИЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ**

Первые проекты электронных вычислительных машин появились в конце 1930-х – начале 1940-х гг. Технические предпосылки для этого уже были созданы. В 1904 г. был изобретен первый ламповый диод, а в 1906 г. – первый триод (соответственно двух- и трехэлектродные электронные лампы); в 1914 г. было изобретено электронное реле (ламповый триггер). Триггерные схемы стали широко применяться в электронике для переключения и релейной коммутации.

Благодаря развитию вычислительной техники в середине 1930-х гг. стало возможным создание программно-управляемых вычислительных машин (ВМ), а построение ВМ на электронных схемах открывало широкие перспективы, связанные с увеличением их надежности и быстродействия.

Компьютеры появились, когда возникла острая необходимость в проведении трудоемких и точных расчетов. Уровень прогресса в таких областях науки и техники, как, например, атомная энергетика и аэрокосмические исследования, во многом зависел от возможности выполнения сложных расчетов, которые нельзя было осуществить на

электромеханических счетных машинах. Требовался переход к вычислительным машинам, работающим с большей производительностью.

Во все времена людям необходимо было считать. Сначала для счета использовали пальцы рук или камешки. Но даже простые арифметические операции с большими числами трудны для мозга человека. Поэтому уже более пятнадцати веков назад в странах Средиземноморья был придуман простейший инструмент для счета – абак, представляющий собой набор костяшек, нанизанных на стержни. Этот прообраз современных счетов широко использовали купцы. Стержни абака в арифметическом смысле представляют собой десятичные разряды. Каждая костяшка на первом стержне имеет достоинство 1, на втором – 10, на третьем – 100 и т. д.

В России так называемые русские счеты – в их основе лежит десятичная система счисления – появились в XVI в. До XVII в. счеты оставались практически единственным счетным инструментом.

В 1614 г. математик Дж. Непер изобрел логарифмы. Открытие Непера состояло в том, что таким способом можно выразить любое число и что сумма логарифмов двух любых чисел равна логарифму произведения этих чисел. Это дало возможность свести действие умножения к более простому действию сложения. Непер создал таблицы логарифмов. Чтобы перемножить два числа, нужно найти в этой таблице их логарифмы, сложить их и отыскать число, соответствующее этой сумме, в обратной таблице – таблице антилогарифмов. На основе этих таблиц в 1654 г. Р. Биссакар и в 1657 г. – независимо от него – С. Партридж разработали прямоугольную логарифмическую линейку – основной счетный прибор инженера до середины XX в.

В 1642 г. Блез Паскаль изобрел механическую суммирующую машину, использующую десятичную систему счисления. Каждый десятичный разряд представляло колесико с десятью зубцами, обозначавшими цифры от 0 до 9. Всего колесиков было восемь, т. е. машина Паскаля была восьмиразрядной.

В 1673 г. Готфрид Лейбниц построил механический арифмометр, выполнявший все четыре арифметических действия. В нем использовалась двоичная система счисления.

В 1804 г. Жозеф Жаккар создал ткацкую машину для выработки тканей с крупным узором. Этот узор программировался с помощью целой колоды перфокарт – прямоугольных карточек из картона. На них информация об узоре записывалась пробивкой отверстий (перфораций), расположенных в определенном порядке. При работе машины эти перфокарты «ощупывались» специальными штырями. Именно таким механическим способом с них считывалась информация для плетения запрограммированного узора ткани. Машина Жаккара явилась прообразом машин с программным управлением, созданных в XX в.

В 1820 г. Тома де Кольмар разработал первый коммерческий арифмометр, способный умножать и делить. Арифмометры получили широкое распространение при выполнении сложных расчетов.

В 1830 г. Чарльз Бэббидж попытался создать универсальную аналитическую машину, которая должна была выполнять вычисления без участия человека. Для этого в нее вводились программы, заранее записанные на перфокарты из плотной бумаги посредством отверстий, сделанных на них в определенном порядке. Принципы программирования для аналитической машины Бэббиджа разработала в 1843 г. Ада Лавлейс – дочь английского поэта Байрона.

Аналитическая машина должна была запоминать данные и промежуточные результаты вычислений, т. е. иметь память. Эта машина содержала три основные части: устройство для хранения чисел, набравшихся с помощью зубчатых колес (память), устройство для операций над числами (арифметическое устройство) и устройство для операций над числами с помощью перфокарт (устройство программного управления).

Работа по созданию аналитической машины не была завершена, но заложенные идеи помогли в XX в. построить первые компьютеры (напомним: в переводе с английского «компьютер» означает «вычислитель»).

В 1880 г. в России Вильголт Однер изобрел механический арифмометр с зубчатыми колесами, а в 1890г. начался его массовый выпуск. Под названием «Феликс» этот арифмометр выпускался вплоть до 1950г.

В 1888 г. Герман Холлерит создал первую электромеханическую счетную машину – табулятор, в котором нанесенная на перфокарты информация расшифровывалась электрическим током. В 1890 г. изобретение Холлерита было впервые использовано во второй американской переписи населения. Работа, которую 500 сотрудников раньше выполняли целых семь лет, Холлерит с 43 помощниками на 43 табуляторах выполнили всего за один месяц.

В 1937 г. Джордж Стибиц из обыкновенных электромеханических реле создал двоичный сумматор – устройство, способное выполнять операцию сложения чисел в двоичном коде. И сегодня двоичный сумматор является одним из главных компонентов любого компьютера, основой его арифметического устройства.

В 1937–1942 гг. Джордж Атанасофф работал над моделью вычислительной машины на вакуумных электронных лампах. В ней использовалась двоичная система счисления. Для ввода данных и вывода результатов вычислений применялись перфокарты. В 1942 г. работа над этой машиной была практически завершена, но из-за войны финансирование исследований прекратилось.

В 1937 г. Конрад Цузе сконструировал на основе электромеханических реле свою первую вычислительную машину – Z1. Исходные данные вводились в нее с помощью клавиатуры, а результат вычислений высвечивался на панели с множеством электрических лампочек. В 1938 г. К. Цузе создал усовершенствованную модель – Z2. Программы в нее вводились с помощью перфоленты. Перфоленту изготавливали, пробивая отверстия в использованной 35-миллиметровой фотопленке. В 1941 г. К. Цузе построил действующий компьютер Z3, а позднее – и Z4, основанный на двоичной системе счисления. Они использовались для расчетов при создании самолетов и ракет. В 1942 г. Конрад Цузе и Хельмут Шрайер задумали перевести Z3 с электромеханических реле на вакуумные электронные лампы. Такая машина должна была работать в тысячу раз быстрее, но создать ее не удалось – помешала война.

В 1943–1944 гг. на одном из предприятий в сотрудничестве с учеными Гарвардского университета во главе с Говардом Эйкеном была создана вычислительная машина «Марк-1». Весила она около 35 т. В «Марк-1» использовались идеи, заложенные Ч. Бэббиджем в его аналитической машине. В отличие от Стибица и Цузе, Эйкен не осознал преимуществ двоичной системы счисления и в своей машине использовал десятичную систему. Машина могла манипулировать числами длиной до 23 разрядов. Для перемножения двух таких чисел ей было необходимо затратить 4 с. В 1947 г. была создана машина «Марк-2», в которой использовалась уже двоичная система счисления. В этой машине операции сложения и вычитания занимали в среднем 0,125 с, а умножение – 0,25 с.

Электромеханические реле работали слишком медленно. Поэтому уже в 1943 г. американцы начали разработку вычислительной машины на основе электронных ламп. В 1946 г. Преспер Эккерт и Джон Мочли построили первую электронную цифровую вычислительную машину – ENIAC. Ее вес составлял 30 т, она занимала площадь 170 м<sup>2</sup>. Вместо огромного числа электромеханических реле, в ENIAC работало 18 тыс. электронных ламп. Считала машина в двоичной системе и производила 5000 операций сложения или 300 операций умножения в секунду. На электронных лампах было построено не только арифметическое, но и запоминающее устройство. Числовые данные

вводились с помощью перфокарт, а программы – с помощью штекеров и наборных плат, т. е. для каждой новой программы приходилось соединять тысячи контактов. Поэтому для подготовки к решению новой задачи требовалось несколько дней, хотя сама задача решалась за несколько минут. Это был один из основных недостатков машины.

### 3.5.2. ПОКОЛЕНИЯ КОМПЬЮТЕРОВ

В истории развития компьютера выделяют пять этапов, соответствующих пяти поколениям ЭВМ.

Период машин *первого поколения* начинается с переходом к серийному производству компьютеров в начале 1950-х гг. В них были реализованы основные принципы, предложенные Джоном фон Нейманом (см. подраздел 1.1.3).

Структура компьютера, в которой реализованы принципы фон Неймана, впоследствии получила название структуры его имени (или классической). Все дальнейшее развитие компьютера шло двумя путями: путём совершенствования структуры фон Неймана и путём поиска новых структур.

Технической основой элементной базы процессоров первого поколения компьютера были электронные вакуумные лампы. Это были громоздкие машины, занимающие много места и потребляющие много электроэнергии. Они делали несколько тысяч операций в секунду и обладали памятью в несколько тысяч машинных слов. Такие машины предполагали монопольный режим использования, т. е. в распоряжении пользователя были все ресурсы машины и ее управление. Программист писал свою программу в машинных кодах и отлаживал ее за пультом машины, которая на время отладки была полностью в его распоряжении. При этом 90 % времени машина простаивала в ожидании команд, т.е. использование машинных ресурсов было малоэффективным из-за отсутствия развитой операционной системы. Использовались компьютеры первого поколения в основном для научных расчетов.

Первой отечественной ЭВМ была малая электронная счетная машина (МЭСМ), разработанная в 1947–1951 гг. под руководством академика С.А. Лебедева. В 1952 г. была введена в эксплуатацию большая электронная счетная машина (БЭСМ), созданная также под его руководством. В 1955 г. начался выпуск малой электронной счетной машины «Урал-1» (руководитель проекта Б.И. Рамеев). Примером зарубежной серийной модели ЭВМ является ИВМ-701 (США).

*Второе поколение* ЭВМ (конец 1950-х – середина 1960-х гг.) называют транзисторно-ферритовым, так как транзисторы (твердые диоды и триоды) заменили электронные лампы в процессорах, а ферритовые (намагничиваемые) сердечники – электронно-лучевые трубки в оперативных запоминающих устройствах.

Применение транзисторов существенно повлияло на характеристики и структуру машин. Транзисторные схемы позволили на порядок повысить плотность монтажа электронной аппаратуры и существенно (на несколько порядков) снизить потребление электроэнергии. Срок службы транзисторов на два-три порядка превосходил срок службы электронных ламп. Скорость компьютера возросла до сотен тысяч операций в секунду, а память – до десятков тысяч машинных слов.

Создание долговременной памяти на магнитных дисках и лентах, а также возможность подключения к компьютерам изменяемого состава внешних устройств существенно расширили функциональные возможности вычислительных машин.

В организации вычислительного процесса также были крупные достижения – это совмещение во времени вычислений и ввода-вывода информации, переход от монопольного режима использования ресурсов машины к пакетной обработке. Задания для компьютера (на перфокартах, магнитных лентах или дисках) собирались в пакет, кото-

рый обрабатывался без перерыва между заданиями. Это позволило более экономно использовать ресурсы машины.

Были разработаны методы программирования в символических обозначениях, созданы первые алгоритмические языки и трансляторы с этих языков, библиотеки стандартных программ.

Наиболее широкое применение нашли отечественные ЭВМ БЭСМ-4, М-220, «Минск-32». Типичным представителем зарубежной ЭВМ второго поколения является IBM-7090.

**Третье поколение** ЭВМ (конец 1960-х – начало 1970-х гг.) характеризуется появлением в качестве элементной базы процессора интегральных полупроводниковых схем (вместо отдельных транзисторов), что привело к дальнейшему увеличению скорости до миллиона операций в секунду и памяти – до сотен тысяч слов.

Компьютеры третьего поколения также характеризуются крупнейшими сдвигами в архитектуре ЭВМ, их программном обеспечении, организации взаимодействия человека с машиной. Это прежде всего наличие развитой конфигурации внешних устройств (алфавитно-цифровые терминалы, графопостроители и т. п.) с использованием стандартных средств сопряжения и развитой операционной системы, обеспечивающей работу в мультипрограммном режиме (несколько одновременно размещаемых в оперативной памяти программ совместно используют ресурсы процессора). Метод использования ресурсов ЭВМ – режим разделения времени совместно с пакетной обработкой. Высокое быстродействие позволяет разбить на кванты время обслуживания пользователей, обрабатывать в течение кванта задание каждого и возвращаться к пользователю за такое малое время, что у него создается иллюзия, будто он один пользуется ресурсами машины.

Решающее значение в развитии вычислительной техники во всем мире сыграло создание семейства вычислительных машин на интегральных схемах с широким диапазоном вычислительной мощности и совместимых снизу вверх на уровне машинных языков, внешних устройств, модулей конструкции и системы элементов. **Программная совместимость** снизу вверх машин одного и того же семейства предполагает, что любая программа, выполнявшаяся на «младшей» машине, должна без всяких переделок выполняться на «старшей».

Широкое распространение получили также семейства мини-ЭВМ. Сущность их конструкторского решения состояла в такой минимизации аппаратуры центрального процессора, которая позволила на уровне технологии того времени создать универсальные ЭВМ, способные осуществлять управление в реальном масштабе времени, при котором темп выдачи управляющих воздействий на объект управления согласован со скоростью протекания процессов в этом объекте.

В нашей стране в период машин третьего поколения была создана Единая система ЭВМ (ЕС ЭВМ), в основных чертах копирующая IBM-360 и IBM-370, а также серия мини-ЭВМ(СМ ЭВМ), ориентированная на зарубежные модели. Вклад отечественной науки в мировое развитие электронной вычислительной техники в этот период связан с промышленным внедрением многопроцессорной ЭВМ М-10.

В период машин третьего поколения произошел крупный сдвиг в области применения ЭВМ. Если раньше они использовались в основном для научно-технических расчетов, то в 1960 – 1970-е гг. первое место стала занимать обработка символьной информации, в основном экономической.

Машины серии ЕС ЭВМ имели универсальное назначение, а основной областью применения СМ ЭВМ была автоматизация технологических процессов, научных экспериментов, испытательных установок и проектно-конструкторских работ.

Переход к **машинам четвертого поколения** – ЭВМ на больших интегральных схемах (БИС) – происходил во второй половине 1970-х гг. и завершился приблизительно к 1980 г. Теперь на одном кристалле размером 1 см<sup>2</sup> стали размещаться сотни тысяч

электронных элементов. Скорость и объем памяти возросли в десятки тысяч раз, по сравнению со скоростью и объемом памяти машин первого поколения, и составили примерно  $10^9$  операций в секунду и  $10^7$  слов соответственно.

Характерные особенности машин четвертого поколения – тесная связь аппаратной и программной реализаций в структуре машины, отход от принципа минимизации аппаратуры и поручение ей функций программы, что стало возможным благодаря относительно низкой стоимости БИС.

Развитие архитектуры ЭВМ в период машин четвертого поколения привело к появлению структур, в которых вычислительный процесс может протекать по нескольким ветвям параллельно, что приводит к увеличению производительности вычислительных машин. Идея параллелизма была технически реализована в многопроцессорных системах, состоящих из двух или более взаимосвязанных процессоров, работающих с общей памятью и управляемых общей операционной системой.

В результате возросшего быстродействия ЭВМ стало возможным расширить оперативную память за счет введения виртуальной памяти, основанной на страничном обмене информацией между внешней и основной памятью.

Наиболее крупным достижением, связанным с применением БИС, стало создание микропроцессоров, а затем на их основе – микроЭВМ. Если прежние поколения ЭВМ требовали для своего размещения специальных помещений, системы вентиляции, специального оборудования для электропитания, то требования, предъявляемые к эксплуатации микроЭВМ, ничем не отличаются от условий эксплуатации бытовых электроприборов. При этом они имеют высокую производительность, экономичны в эксплуатации и дешевы. МикроЭВМ используются в измерительных комплексах, системах числового программного управления, в управляющих системах различного назначения.

Дальнейшее развитие микроЭВМ привело к созданию персональных компьютеров, широкое распространение которых началось с 1975 г., когда фирма IBM выпустила свой первый персональный компьютер IBM PC. Сейчас такие компьютеры (совместимые с IBM PC) составляют около 90 % всех производимых в мире ПК. В персональных компьютерах реализован принцип *открытой архитектуры*, который означает, что по мере улучшения характеристик основных блоков ПК можно легко заменить устаревшие части, а модернизированный блок будет совместим с ранее используемым оборудованием. В числе других преимуществ ПК – развитые средства диалога, высокая надежность, удобство эксплуатации, наличие программного обеспечения, охватывающего практически все сферы человеческой деятельности.

Начали серийно производиться и суперЭВМ. Рост степени интеграции БИС стал технологической основой производительности ЭВМ. В нескольких серийных моделях была достигнута производительность свыше 1 млрд операций в секунду. К числу наиболее значительных разработок машин четвертого поколения относится ЭВМ «Крей-3» (США), спроектированная на основе принципиально новой технологии – замены кремниевого кристалла арсенидом галлия и имеющая производительность до 16 млрд операций в секунду. Примером отечественной суперЭВМ является многопроцессорный вычислительный комплекс «Эльбрус» с быстродействием до  $1,2 \times 10^8$  операций в секунду.

С конца 1980-х гг. в соответствии с идеологией развития компьютерных технологий, после четвертого поколения ожидалось создание компьютера следующего *пятого поколения*, ориентированного на распределенные вычисления и способного к имитации мышления. Программа закончилась провалом, так как не опиралась на четкие научные методики, более того, даже её промежуточные цели оказались недостижимы в технологическом плане.

В настоящий момент термин «пятое поколение» является неопределенным и применяется во многих смыслах, например при описании систем облачных вычислений.

### 3.5.3. ПЕРСПЕКТИВЫ РАЗВИТИЯ ПК

#### Понятие ПК

Персональный компьютер – компьютер, предназначенный для эксплуатации одним пользователем, то есть для личного использования. Хотя изначально компьютер был создан как вычислительная машина, в качестве ПК он обычно используется в других целях – как средство доступа в информационные сети и как платформа для компьютерных игр.

В употребление термин был введён в конце 1970-х годов компанией Apple Computer для своего компьютера Apple II и впоследствии перенесён на компьютеры IBM PC. Некоторое время персональным компьютером называли любую машину, использующую процессоры Intel и работающую под управлением операционных систем DOS, OS/2 и первых версий Microsoft Windows. С появлением других процессоров, поддерживающих работу перечисленных программ, таких, как AMD, Cyrix (ныне VIA), название стало иметь более широкую трактовку. Курьёзным фактом стало противопоставление «персональным компьютерам» вычислительных машин Amiga и Macintosh, долгое время использовавших альтернативную компьютерную архитектуру.

Чаще всего под ПК понимают настольные ПК, ноутбуки, планшетные и карманные ПК. Однако на самом деле персональным может считаться любой полноценный компьютер – даже суперкомпьютер – используемый в качестве персонального, то есть личного, компьютера.

В Советском Союзе вычислительные машины, предназначенные для личного использования, носили официальное название «персональные электронные вычислительные машины» (ПЭВМ). В терминологии, принятой в российских стандартах, это словосочетание и сегодня указывается вместо используемого де-факто названия «персональный компьютер».

#### Стационарные ПК

Первые персональные компьютеры (как и любые первые компьютеры вообще) не предназначались для переноски. То есть первые ПК были стационарными. Они состояли из отдельных конструктивно завершённых частей: системного блока, монитора, клавиатуры. Это пример раздельной схемы построения ПК. Но в настоящее время также широкое распространение получили ПК-моноблоки, в которых системный блок, монитор и нередко другие устройства (клавиатура, звуковая подсистема, веб-камера, микрофон) конструктивно объединены в одно устройство.

Раздельная схема в противоположность моноблочной предполагает, что ПК состоит из системного блока и разнообразных внешних, то есть конструктивно самостоятельных подключаемых к системному блоку извне через стандартные интерфейсы (например: USB, D-Sub, DVI, FireWire), устройств (монитора, клавиатуры, мыши, микрофона, звуковых колонок, веб-камеры, принтера, сканера, различных внешних модемов, игровых устройств). Исторически такая схема ПК было самой первой. Она же до сих пор остается самой распространенной схемой стационарных ПК. Например, профессиональные рабочие станции практически всегда строятся по такой схеме.

Главное достоинство раздельной схемы – сравнительно легкая масштабируемость. То есть в любой момент можно без особых затруднений заменить любой из компонентов ПК (например, монитор). Недостатком является меньшая транспортабельность и сравнительная громоздкость такого ПК. Естественно раздельная схема применяется тогда когда главное требование к ПК – легкость и простота масштабирования.

Функциональным ядром в раздельной схеме стационарного ПК, естественно, является системный блок. Известны два вида конструктивной компоновки системного блока:

- **desktop** – горизонтальная конструктивная компоновка системного блока, с возможностью размещения монитора на таком системном блоке;

- *tower* – «башенный» системный блок в вертикальной конструктивной компоновке.

Десктоп («настольный компьютер» в буквальном смысле слова) – стационарный компьютер, имеющий такой форм-фактор, что его удобнее располагать на столе (отсюда и применение термина «десктоп», от англ. *desktop* – «рабочая поверхность (письменного стола)») дома или в офисе. Раньше системные блоки такого типа обычно были широкими и места на них было достаточно для размещения на нём электронно-лучевого монитора, что, в свою очередь, позволяло экономить место на рабочем столе, на который устанавливался десктоп. Естественно, это было учтено конструкторами корпусов, создававшими корпуса для таких системных блоков, способные выдерживать вес ЭЛТ-монитора. Но в результате стоимость десктопа получалась больше, чем стоимость «башенного» системного блока.

Десктопы применяются до сих пор, и до сих пор монитор ставят на десктоп. Однако из-за уменьшения габаритов и веса комплектующих и ещё более резкого уменьшения веса и глубины мониторов (современные «доскообразные» мониторы – сплошь жидкокристаллических мониторов – сравнительно малы по весу и глубине), стало возможным создавать и использовать сравнительно компактные и дешёвые десктопы. В частности многими фирмами выпускаются тонкие десктопы – слим-десктопы (*slim-desktop*). Естественно, тонкий десктоп эргономичнее чем классический «толстый» десктоп, так как почти не влияет на высоту установки размещаемого на нём монитора.

«Башенный» системный блок – системный блок типа Tower («башня») – высокий и потому обычно располагается под столом (часто в специально предназначенных для это нишах или отделениях компьютерных столов). Из-за уменьшения размеров и массы комплектующих также стало возможно уменьшение размеров самих «башенных» системных блоков. В результате сначала появились системные блоки *minitower*, а потом и *slim-tower*. *Minitower*, правда, потом вышли из эксплуатации, уступив свое место системным блокам *middletower*, являющихся в настоящее время самой многочисленной подгруппой «башенных» системных блоков. А вот *slim-tower* безраздельно господствует в категории компактных «башенных» системных блоков.

Конструктивная схема стационарного ПК, в которой системный блок, монитор и в настоящее время микрофон, звуковая колонки, веб-камера конструктивно объединены в одно устройство – *моноблок*. Такой ПК эргономичнее (занимает минимум пространства) и более привлекателен с эстетической точки зрения. Он и более транспортабелен чем стационарный ПК, построенные по отдельной схеме. Недостатками является сравнительно трудная масштабируемость такого ПК, в том числе самостоятельная техническая модернизация, а также сложное техническое обслуживание. Если у моноблока сломается, например, микрофон, то заменить его на исправный нередко возможно только в сервис-центре. Естественно, моноблоки широко применяются в тех случаях, когда на первом месте стоит экономичность, а простота масштабирования или возможность самостоятельного технического обслуживания, напротив, не являются решающими.

### Ноутбук

Ноутбук – компактный компьютер, содержащий все необходимые компоненты (в том числе монитор) в одном небольшом корпусе, как правило, складывающемся в виде книжки (отсюда и название данного вида ПК). Приспособлены для работы в дороге, на небольшом свободном пространстве. Для достижения малых размеров в них применяются специальные технологии: специально разработанные специализированные микросхемы (ASIC), ОЗУ и жёсткие диски уменьшенных габаритов, компактная клавиатура, не содержащая цифрового поля, внешние блоки питания, минимум интерфейсных гнезд для подключения внешних устройств.

Ноутбуки, как правило, содержат развитые средства подключения к проводным и беспроводным сетям, встроенное мультимедийное оборудование (динамики, микрофон, веб-

камеру). В последнее время вычислительная мощность и функциональность ноутбуков не сильно уступают мощности и функциональности стационарных ПК, а иногда и превосходит их. Очень компактные модели ноутбуков не оснащаются встроенным CD/DVD-дисководом.

Подключая к ноутбуку внешние клавиатуру, мышь, монитор, звуковые колонки, модемы, игровые устройства и иные внешние устройства ноутбук можно превратить настольный ПК. Это можно делать, вставляя ноутбук в специальный док, как это делалось раньше или напрямую (благо современные ноутбуки, особенно предназначенные для замены стационарных ПК в качестве рабочих станций, дают такую возможность).

### **Планшетные ПК**

Планшетные ПК аналогичны ноутбукам, но содержат сенсорный, то есть чувствительный к нажатию, экран и не содержат механической клавиатуры. Ввод текста и управление осуществляются через экранный интерфейс, часто доработанный специально для удобного управления пальцами. Некоторые модели могут распознавать рукописный текст, написанный на экране.

Чаще всего корпус планшетных ПК не раскрывается, как у ноутбуков, а экран расположен на внешней стороне верхней поверхности. Бывают и комбинированные модели, у которых корпус может тем или иным образом раскрываться (например как слайдер), предоставляя доступ к расположенной внутри клавиатуре.

По вычислительной мощи планшетные ПК уступают стационарным ПК и ноутбукам, так как для длительной работы без внешнего источника питания приходится использовать энергосберегающие комплектующие, жертвуя их быстродействием.

### **Карманные ПК (PDA)**

Карманные ПК (PDA) – сверхпортативные ПК, уместяющиеся в кармане (в России их также часто называют «наладонниками», «карманками»). Управление ими, как правило, происходит с помощью небольшого по размерам и разрешению экрана, чувствительного к нажатию пальца или специальной палочки-указки – стилуса, а клавиатура и мышь отсутствуют. Некоторые модели, впрочем, содержат миниатюрную фиксированную или выдвигающуюся из корпуса клавиатуру. Разрешение экрана стремится приблизиться к разрешению мониторов обычных компьютеров.

В карманных ПК (КПК) используются сверхэкономичные процессоры и флеш-накопители небольшого объёма, поэтому по вычислительной мощи они несопоставимы с другими ПК (особенно стационарными). Тем не менее, они содержат все признаки персонального компьютера: процессор, накопитель, оперативную память, монитор, операционную систему, прикладное ПО и даже игры и ориентированность на индивидуальное использование.

Всё более популярными становятся КПК, содержащие также функции мобильного телефона (коммуникаторы). Встроенный коммуникационный модуль позволяет не только совершать звонки, но и подключаться к интернету в любой точке, где есть сотовая связь совместимого стандарта.

### **Varebone**

Varebone – компьютеры, строящиеся пользователем для выполнения определенных задач (обычно в качестве мультимедийной станции). В продажу поступают в виде так называемых «скелетных» баз в составе корпуса, материнской платы и системы охлаждения. Материнская плата, как правило, оснащена встроенными звуковым и видеоконтроллерами. Выбор конфигурации и соответственно комплектующих в виде дисковых накопителей, памяти и периферии, а также других устройств (ТВ-тюнера, дополнительной видеокарты и т. п.) остаётся на усмотрение пользователя. Как правило, «барэбоны» имеют меньшую высоту

корпуса и, как следствие, уменьшенный внутренний объём, а также усовершенствованную систему охлаждения, отличающуюся низкой шумностью.

### **Защищённые ПК**

Ряд компаний производит компьютеры, обладающие устойчивостью к агрессивным средам: сильной вибрации, ударам, большой запыленности, влажности, вандализму – условиям, в которых обычные ПК быстро бы вышли из строя. Как правило, устойчивые ПК выпускаются в формате ноутбуков, более тяжёлых и больших по размерам, чем обычные. Их стоимость значительно выше, чем стоимость обычных ПК. Одна из сфер применения таких ПК – военное дело (например, эксплуатация в полевых штабах).

### **Промышленные ПК**

Промышленные ПК предназначены для решения задач промышленной автоматизации. Отличаются стойкостью к различным внешним воздействиям, увеличенным жизненным циклом изделия, возможностью подключения к промышленным сетям.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Дайте определение компьютера.
2. Сформулируйте принципы построения компьютера фон Неймана.
3. Что такое архитектура и конфигурация компьютера?
4. Опишите шинную организацию компьютера.
5. Назовите состав и функции блоков центрального процессора компьютера.
6. Для чего предназначены регистры?
7. Какие функции выполняет устройство управления компьютера?
8. Расскажите о составе устройств памяти компьютера.
9. Какова система шин компьютера?
10. Какова последовательность работы элементов компьютера при обработке информации?
11. Что такое персональный компьютер?
12. В чем отличие ПК от ЭВМ других классов?
13. Из каких устройств состоит системный блок? Каково их назначение?
14. Какие периферийные устройства входят в состав ПК?
15. Каковы назначение и основные характеристики периферийных устройств?
16. Расскажите об известных вам внешних запоминающих устройствах ПК.
17. Какие виды печатающих устройств ПК вы знаете?
18. Что такое операционная система ПК?
19. Дайте определение понятия «пользовательский интерфейс».
20. Из каких этапов состоит технология информационных процессов в ПК?
21. Назовите функции текстовых процессоров.
22. Дайте краткую характеристику табличного процессора.
23. Как вы понимаете диалоговый режим работы ПК?
24. Перечислите основные направления развития ПК.
25. По каким характеристикам осуществляется классификация ЭВМ по поколениям?
26. Дайте характеристику ЭВМ четвертого поколения.
27. Расскажите об особенностях ЭВМ пятого поколения.

## 4. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

### 4.1. КЛАССИФИКАЦИЯ И ХАРАКТЕРИСТИКА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

#### 4.1.1. ПОНЯТИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

*Информационная технология* – это промышленные способы обработки информации.

Пакеты прикладных программ делают применение компьютера массовым, освобождая пользователя от самостоятельного составления сложных программ. Программным средством можно пользоваться, даже не зная его структуры и методов, положенных в его основу. С помощью одного и того же ППП можно решать различные классы задач (иногда не очень похожих друг на друга). Информационные технологии существенно облегчают хранение и поиск нужной информации. Для решения различных жизненных задач нужны разные ИТ. Ресурсы компьютера (в первую очередь объем памяти и быстродействие) и их эксплуатационные характеристики влияют на возможности, предоставляемые ИТ.

Необходимо подчеркнуть сложность технологического процесса сбора, передачи, обработки и хранения информации и важность информационных систем как составляющей инфраструктуры общества, в котором информация выступает одним из главных ресурсов его жизнедеятельности. Автоматизированная информационная технология представляет собой совокупность методов и способов сбора, передачи, накопления, хранения, поиска и обработки информации на основе применения средств вычислительной техники и связи.

Любая вычислительная сеть – это совокупность технических, коммуникационных и программных средств, обеспечивающих эффективное распределение вычислительных ресурсов. Вычислительные сети, являясь одновременно и продуктом, и мощным стимулом развития интеллекта человека, позволяют:

- построить мощные хранилища информации (БД);
- расширить перечень решаемых задач по обработке информации;
- повысить надежность информационной системы за счет дублирования работы ПК;
- создать новые виды сервисного обслуживания, например электронную почту;
- снизить стоимость обработки информации.

Появление новейших ИТ в среде глобальных сетей открыло дорогу к централизации обработки и использования информации.

Современные информационные системы организационного управления предназначены оказывать помощь специалистам и руководителям, принимающим решения, в получении в необходимом количестве своевременной и достоверной информации; создании условий для организации автоматизированных офисов, проведения с применением компьютеров и средств связи оперативных совещаний, сопровождаемых звуковым и видеорядом. Достигается это переходом на новую информационную технологию.

Новая (современная) информационная технология – это технология, основанная на следующем:

- повсеместном применении ЭВМ и оргтехники;
- активном участии пользователей (непрофессионалов в области вычислительной техники и программирования) в информационном процессе;
- высоком уровне дружественного пользовательского интерфейса;
- широком применении ППП общего и проблемного назначения;

- возможности доступа пользователя к базам данных и программ, в том числе и удаленным, благодаря локальным и глобальным сетям;
- анализе ситуаций при выработке и принятии управленческих решений с помощью автоматизированных рабочих мест специалистов;
- применении систем искусственного интеллекта;
- внедрении экспертных систем;
- использовании телекоммуникаций;
- создании геоинформационных систем и других технологий.

#### 4.1.2. СОДЕРЖАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Стремительное развитие и совершенствование эксплуатационных возможностей ЭВМ создало предпосылки для автоматизации умственного труда и формирования рынка информационных продуктов и услуг. Развитие ИТ шло параллельно с появлением новых видов технических средств обработки и передачи информации, новых средств коммуникаций.

В настоящее время ИТ классифицируют по следующим признакам:

- способу реализации в автоматизированных информационных системах (АИС);
- степени охвата задач управления;
- классам реализуемых технологических операций;
- типу пользовательского интерфейса;
- вариантам использования сети;
- обслуживаемым предметным областям;
- др.

По способу реализации в АИС информационные технологии подразделяют на традиционные и современные. **Традиционные ИТ** существовали в условиях централизованной обработки данных, до периода массового использования ПК. Они были ориентированы главным образом на снижение трудоемкости пользователя (например, инженерные и научные расчеты, формирование регулярной отчетности на предприятиях и др.). **Новые (современные) ИТ** связаны в первую очередь с информационным обеспечением процесса управления в режиме реального времени.

По степени охвата задач управления выделяют ИТ:

- электронной обработки данных;
- автоматизации функций управления;
- поддержки принятия решений;
- электронного офиса;
- экспертной поддержки.

**Электронная обработка данных** выполняется с использованием ЭВМ без пересмотра методологии и организации процессов управления при решении локальных математических и экономических задач. При **автоматизации управленческой деятельности** вычислительные средства, включая суперЭВМ и ПК, используются для комплексного решения функциональных задач, формирования регулярной отчетности и работы в информационно-справочном режиме, для подготовки управленческих решений. К этой же группе относятся ИТ **поддержки принятия решений**, которые предусматривают широкое использование экономико-математических методов и моделей, ППП для аналитической работы и формирования прогнозов, составления бизнес-планов, обоснованных оценок и выводов относительно процессов и явлений производственно-хозяйственной деятельности.

К названной группе относятся и широко внедряемые в настоящее время ИТ *электронного офиса* и *экспертной поддержки* решений. Эти два варианта ИТ ориентированы на использование достижений в области новейших методов автоматизации работы специалистов и руководителей, создание наиболее благоприятных условий для выполнения профессиональных функций, качественного и своевременного информационного обслуживания за счет автоматизированного набора управленческих процедур, реализуемых в условиях конкретного рабочего места и офиса в целом.

Электронный офис предусматривает наличие интегрированных ППП, которые обеспечивают комплексную реализацию задач предметной области. В настоящее время все большее распространение приобретают электронные офисы, сотрудники и оборудование которых могут находиться в разных помещениях. Необходимость работы с документами, материалами и БД конкретного предприятия или учреждения в гостинице, транспорте, дома привела к появлению электронных офисов, включенных в соответствующие сети.

Информационные технологии экспертной поддержки составляют основу автоматизации труда специалистов-аналитиков. Эти работники, кроме аналитических методов и моделей для исследования складывающихся ситуаций, вынуждены использовать сведения, составляющие базу знаний в конкретной предметной области. По классам реализуемых технологических операций информационные технологии подразделяют на ИТ работающие:

- с текстовым и табличным процессорами;
- с графическими объектами;
- с системами управления БД;
- с гипертекстовыми и мультимедийными системами.

Подчеркнем, что создание программных средств для вывода высококачественного звука и видеоизображения – перспективное направление развития компьютерной технологии. Технология формирования видеоизображения получила название компьютерной графики. Компьютерная графика – это создание, хранение и обработка моделей объектов и их изображений с помощью ЭВМ. Эта технология проникла в область моделирования различных конструкций (в машиностроении, авиационной технике, автомобилестроении, строительной технике и др.), экономического анализа, проникает в рекламную деятельность, делает занимательным досуг.

Формируемые и обрабатываемые с помощью цифрового процессора изображения могут быть демонстрационными и анимационными. К демонстрационным изображениям относят, как правило, коммерческую (деловую) и иллюстративную графику. К анимационной графике принадлежит инженерная и научная графика, а также графика, связанная с рекламой, искусством, играми, когда на экран выводятся не только одиночные изображения, но и последовательность кадров в виде фильма (интерактивный вариант).

Интерактивная графика – одно из наиболее прогрессивных направлений среди современных ИТ. Бурное развитие этого направления связано с появлением новых графических станций и специализированных программных средств, позволяющих создавать реалистичские объемные движущиеся изображения, сравнимые по качеству с кадрами видеофильма.

В классическом понимании система управления БД представляет собой набор программ, позволяющих создавать и поддерживать БД в актуальном состоянии. Обычно любой текст представляется как одна длинная страница символов, которая читается в одном направлении. Гипертекстовая технология заключается в том, что текст представляется как многомерный, т. е. как иерархическая структура. Материал текста делится на фрагменты. Каждый видимый на экране ЭВМ фрагмент, дополненный многочисленными связями с другими фрагментами, позволяет уточнить информацию об изучаемом объекте и двигаться в одном или нескольких направлениях по выбранной связи.

Программно-техническая организация обмена с компьютером текстовой, графической, аудио- и видеoinформацией получила название *мультимедиа-технологии*.

По типу пользовательского интерфейса можно рассматривать ИТ с точки зрения возможностей доступа пользователя к информационным и вычислительным ресурсам. Так, пакетная ИТ исключает возможность пользователя влиять на обработку информации, пока она проводится в автоматическом режиме. Это объясняется организацией обработки информации, которая основана на выполнении программно-заданной последовательности операций над заранее накопленными в системе и объединенными в пакет данными. В отличие от пакетной, диалоговая ИТ предоставляет пользователю неограниченную возможность взаимодействовать с хранящимися в системе информационными ресурсами в реальном масштабе времени, получая при этом всю необходимую информацию для решения функциональных задач и принятия решений. Интерфейс сетевой ИТ предоставляет пользователю средства теледоступа к территориально распределенным информационными вычислительным ресурсам благодаря развитым средствам связи.

В настоящее время наблюдается тенденция к объединению ИТ различных типов в единый компьютерно-технологический комплекс, который носит название интегрированного. Особое место в нем принадлежит средствам коммуникации, не только обеспечивающим чрезвычайно широкие технологические возможности автоматизации управленческой деятельности, но и являющимся основой создания самых разнообразных сетевых вариантов ИТ: локальных, многоуровневых, распределенных и глобальных информационно-вычислительных сетей.

Чрезвычайно разнообразна классификация ИТ по обслуживаемым предметным областям. Например, только в экономике таковыми являются бухгалтерский учет, банковская, налоговая и страховая деятельность и др.

## 4.2. ТЕХНОЛОГИЯ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ

### 4.2.1. ТЕКСТОВЫЙ РЕДАКТОР: НАЗНАЧЕНИЕ И ОСНОВНЫЕ ФУНКЦИИ

Пользователь ПК часто встречается с необходимостью подготовки тех или иных документов: писем, статей, служебных записок, отчетов, рекламных материалов и т.д. Для их подготовки текст редактируемого документа выводится на экран, и пользователь может в диалоговом режиме вносить в него свои изменения. Все внесенные изменения фиксируются. При распечатке выводится отформатированный текст, в котором учтены все исправления. Пользователь может переносить части текста из одного места документа в другое, использовать несколько видов шрифтов для выделения отдельных участков текста, печатать подготовленный документ на принтере в нужном количестве экземпляров.

Удобство и эффективность применения компьютера привели к созданию множества программ для обработки документов. Такие программы называются *текстовыми процессорами* или редакторами текстов программ. Другими словами, редактор текста – это программа или ее часть, обеспечивающая создание текстовых документов и их редактирование. При этом текстовый документ – это не только статья, письмо, служебная записка, но и тексты программ, команды ОС. Любой пользовательский интерфейс должен обеспечить работу с текстом, а значит, должен иметь в своем составе текстовый редактор. Около 80 % времени работы всех компьютеров в мире уходит на работу с текстами.

В информатике под документом (от лат. *documentum* – свидетельство) понимается совокупность данных в электронном виде, подтверждающих, объясняющих, свидетельствующих о чем-либо. Это не обязательно текст, это может быть письмо со встроенным музыкальным фрагментом или видеоклипком, доклад со встроенной электронной таблицей, файл презентации.

Возможности текстовых редакторов различны – от программ, предназначенных для подготовки небольших документов простой структуры, до программ для набора, оформления и полной подготовки к типографскому изданию книг и журналов (издательские системы).

Редакторы текстов программ редактируют программы на том или ином языке программирования. Часто они встроены в систему программирования. Они выполняют следующие функции:

- диалогового просмотра текста;
- редактирования строк программы;
- копирования и переноса блоков текста;
- копирования одной программы или ее части в указанное место другой программы;
- контекстного поиска и замены текста;
- автоматического поиска строки, содержащей ошибку;
- распечатки программы или ее части.

Часто редакторы текстов программ позволяют автоматически проверять синтаксическую правильность программ; иногда они объединены с отладчиками программ на уровне исходного текста. Редакторы текстов программ можно использовать для создания и корректирования небольших документов. Однако для серьезной работы с документами предпочтительнее использовать специальные редакторы, ориентированные на работу с текстами, имеющими структуру документа, т.е. состоящими из разделов, страниц, абзацев, приложений, слов и т.д.

Редакторы текстов программ обеспечивают следующие функции:

- использования различных шрифтов символов;
- работу с пропорциональными шрифтами;
- задания произвольных междустрочных промежутков;
- автоматического переноса слов на новую строку;
- автоматической нумерации страниц;
- обработки и нумерации сносок;
- печати верхних и нижних заголовков страниц;
- выравнивания краев абзацев;
- набора текста в несколько столбцов;
- проверки правописания и подбора синонимов;
- построения оглавления индексов;
- сортировки текстов и данных и т.д.

Существует несколько сотен редакторов текстов – от самых простых до весьма мощных и сложных. Самый распространенный из них – Microsoft Word.

#### 4.2.2. ИЗДАТЕЛЬСКИЕ СИСТЕМЫ

Для подготовки рекламных буклетов, оформления журналов и книг служат специальные издательские системы. Они позволяют готовить их и печатать на лазерных принтерах или выводить на фотонаборные автоматы сложные документы высокого качества.

**Настольные издательские системы** (НИС) – это программы, предназначенные для профессиональной издательской деятельности, позволяющие осуществлять электронную верстку широкого спектра основных типов документов (информационного бюллетеня, краткой цветной брошюры, объемного каталога, торговой заявки, справочника). Предусмотренные в пакетах данного типа средства позволяют:

- компоновать (верстать) текст;
- использовать всевозможные шрифты и полиграфические изображения;

- осуществлять редактирование текста на уровне лучших текстовых процессоров;
- обрабатывать графические изображения;
- обеспечивать вывод документов высокого качества;
- работать в сетях и на разных платформах.

Наилучшими пакетами в этой области для являются QuarkXPress и Adobe In Design.

Основная операция издательских систем – верстка (размещение текста по страницам документа, вставка рисунков, оформление текста различными шрифтами и т. д.). В режиме ввода редактирования текста они уступают в скорости и удобстве текстовым редакторам. Поэтому чаще всего документы подготавливают в два этапа: набирают текст в редакторе типа Microsoft Word, а затем импортируют его в системы верстки и осуществляют окончательную подготовку документа.

Основными функциями настольных издательских систем являются:

- использование сотен видов шрифтов (начертаний и размеров символов текста), которые отображаются на экране так же, как при печати;
- размещение фрагментов в документе;
- изменение и корректировка рисунков и диаграмм; изменение интервала между буквами (обычный, разреженный или уплотненный);
- формирование таблиц;
- выравнивание нижнего края текста на странице на заданную границу (чтобы страницы документа имели единообразный вид);
- набор формул и т.д.

Многим пользователям для выполнения издательских работ вполне достаточно возможностей Microsoft Word. В последнее время производители издательских систем стали встраивать в них элементы профессионального цветоделения, обеспечивающие подготовку высококачественных цветных изданий, а также средства графических редакторов.

### 4.3. ТЕХНОЛОГИЯ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

#### 4.3.1. ВИДЫ ГРАФИКИ

Графика – наиболее общий способ визуального представления данных в компьютере, в котором объединяются текстовые данные и графические образы.

**Графический редактор** – это программный комплекс, обеспечивающий пользователя средствами для создания графических образов, картин, рисунков и даже мультипликации. Кроме того, графический редактор заботится о сохранности созданного изображения и удобстве работы с принтером, памятью и т. п.

Имеется два способа, или формата, представления графического изображения на машинных носителях – растровая и векторная графика.

**Растровая графика** – способ представления изображения в виде набора точек. **Векторная графика** – это способ представления изображения как совокупности графических элементов (графических примитивов: отрезков, дуг и пр.), описанных любым способом, в том числе графическими командами.

#### 4.3.2. ГРАФИЧЕСКИЕ РЕДАКТОРЫ

Потребность ввода графиков, схем, диаграмм, рисунков в текст или документ вызвала необходимость создания графических процессоров. Графические процессоры представляют собой инструментальные средства, позволяющие создавать и модифицировать графические образы с использованием иллюстративной, коммерческой, научной и когнитивной графики.

Информационные технологии иллюстративной графики позволяют создавать иллюстрации для различных текстовых документов в виде регулярных структур – различных геометрических фигур (векторная графика). Процессоры, реализующие ИТ иллюстративной растровой графики, дают возможность пользователю выбрать толщину и цвет линий, палитру заливки, шрифт для записи и наложения текста, создавать разные графические образы, а также стирать или разрезать рисунок, перемещать его части.

Информационные технологии коммерческой, или деловой, графики обеспечивают отображение информации, хранящейся в табличных процессорах, БД и отдельных локальных файлах, в виде двух- или трехмерных графиков, круговой диаграммы, столбиковой гистограммы, линейных графиков и др. (Excel и т.д.).

Информационные технологии научной графики предназначены для обслуживания задач оформления научных отчетов, содержащих математические, химические и прочие формулы, задач картографии и др.

Графические редакторы-пакеты, предназначенные для обработки графической информации, подразделяют на ППП обработки для растровой и векторной графики.

Пакеты прикладных программ для обработки растровой графики предназначены для работы с фотографиями и включают в себя набор средств по кодированию фотоизображений в цифровую форму. Признанный лидер среди пакетов данного класса – Adobe Photoshop.

Пакеты прикладных программ для обработки векторной графики предназначены для профессиональной работы, связанной с художественной и технической иллюстрацией, с последующей цветной печатью (на рабочем месте дизайнеров, например). Они занимают промежуточное положение между пакетами для систем автоматизированного проектирования и настольными издательскими системами (НИС). Своеобразным стандартом среди пакетов этого класса является CorelDRAW.

Пакеты прикладных программ демонстрационной графики являются конструкторами графических образов деловой информации, призванными в наглядной и динамичной форме представлять результаты некоторого аналитического исследования. Работа с пакетами этого типа строится по следующей схеме:

- 1) разработка общего плана представления;
- 2) выбор шаблона для оформления элементов;
- 3) формирование и импорт элементов, таких, как текст, графики, таблицы, диаграммы, звуковые эффекты и видеоклипы.

К наиболее популярным пакетам данного типа относятся PowerPoint, Harvard Graphics.

#### 4.4. ТЕХНОЛОГИЯ ОБРАБОТКИ ЧИСЛОВОЙ ИНФОРМАЦИИ

Множество задач, которые решают фирмы и предприятия, носят учетно-аналитический характер и требуют табличной компоновки данных с подведением итогов по различным группам и разделам данных, например при составлении баланса, справок для налоговых органов, всевозможных финансовых отчетов и т. п. Для хранения и обработки информации, представленной в табличной форме, используют *электронные таблицы* (ЭТ).

Программные средства для проектирования таблиц называют также табличными процессорами. Они позволяют не только создавать таблицы, но и автоматизировать обработку табличных данных. Кроме того, с помощью ЭТ можно выполнять различные экономические, бухгалтерские и инженерные расчеты, а также строить разного рода диаграммы, проводить сложный экономический анализ, моделировать и оптимизировать решение различных хозяйственных операций и многое другое.

Функции табличных процессоров весьма разнообразны и включают:

- создание и редактирование ЭТ;

- оформление и печать ЭТ;
- создание многотабличных документов, объединенных формулами;
- построение диаграмм, их модификацию и решение экономических задач графическими методами;
- работу с электронными таблицами как с базами данных (сортировку таблиц, выборку данных по запросам);
- создание итоговых и сводных таблиц;
- использование при построении таблиц информации из внешних баз данных;
- решение экономических задач типа «что будет, если...» путем подбора параметров;
- решение оптимизационных задач;
- статистическую обработку данных;
- разработку макрокоманд, настройку среды под потребности пользователя и т.д.

Табличные процессоры различаются в основном набором выполняемых функций и удобством интерфейса, поэтому целесообразно проанализировать лишь широко используемые программные продукты.

#### 4.5. ТЕХНОЛОГИЯ ХРАНЕНИЯ, ПОИСКА И СОРТИРОВКИ ИНФОРМАЦИИ

##### 4.5.1. СПОСОБЫ ОРГАНИЗАЦИИ БАЗ ДАННЫХ

Центральным техническим вопросом разработки АСУ и любых других информационных систем является вопрос организации, хранения и комплексного использования данных. Поэтому получили развитие средства управления данными, которые служат основой любой информационной системы, построенной на базе использования средств вычислительной техники.

Автоматизированные системы управления, спроектированные на основе концепции банков данных, обладают целым рядом характерных свойств, которые выгодно отличают их от предшествующих систем, основой которых были массивы данных. Использование автоматизированных банков данных позволяет обеспечить многоаспектный доступ к совокупности взаимосвязанных данных, высокую степень независимости прикладных программ от изменений логической и физической организации данных, интеграцию и централизацию управления данными, устранение излишней избыточности данных, возможность совмещения пакетов и телепроцессорной обработки данных. Поэтому разработка АСУ для любой сферы применения связана прежде всего с созданием автоматизированных банков данных.

Так как основой любого управления является информация о состоянии объекта, то именно поэтому центром АСУ являются данные, их организация, тщательное ведение, хранение и использование. Меняются техника, программное хозяйство, но данные остаются. Работа с ними – дело дорогое, поэтому ученые задумались над системными принципами организации, положенными в основу создания банков данных.

Под **автоматизированным банком данных** понимается организационно-техническая система, представляющая собой совокупность баз данных пользователей, технических и программных средств формирования и ведения этих баз и коллектива специалистов, обеспечивающих функционирование системы.

Составными частями любого банка данных являются база данных, **система управления базой данных**, администратор базы данных, прикладное программное обеспечение.

Функционирование СУБД основано на введении двух уровней организации базы данных – **логического** и **физического**, соответствующих двум аспектам организации данных – логическому (с точки зрения использования данных в приложениях) и физическому

му (с точки зрения хранения данных в памяти компьютера). Описание логической организации базы данных определяет взгляд пользователей на организацию данных в системе, которые отображают состояние некоторой предметной области. В общем случае структуры физической и логической организации данных могут не совпадать. Формальное описание логической организации данных иногда называют моделью данных или схемой.

Говоря о физической организации, необходимо отметить, что существует много различных способов организации данных в запоминающей среде, с помощью которых можно обеспечить соответствие некоторой модели.

Наиболее общее представление о базе данных можно сформулировать так: база данных представляет собой совокупность хранимых во внешней памяти компьютера больших объемов данных и является интегрированной, т.е. представляет собой комплекс взаимосвязанных данных, предназначенных для обеспечения информационных нужд различных пользователей, каждый из которых имеет отношение к отдельным (возможно, совместно используемым) частям данных. Работа с БД может осуществляться либо в пакетном режиме, либо с удаленных терминалов в режиме реального времени.

Таким образом, *база данных* – это совокупность хранимых в памяти компьютера и специальным образом организованных взаимосвязанных данных, отображающих состояние предметной области. База данных также предназначена для обеспечения информационных нужд определенных пользователей.

База данных хранит информацию об объектах реального мира и отношениях между этими объектами через данные и связи между данными. Прежде чем говорить о размещении данных и связей между ними на устройствах памяти, необходимо представить взаимосвязь данных на логическом уровне, создав своеобразную *модель данных*. Эта модель данных должна быть четко определена, для чего в системе управления базами данных необходимо представить средства (язык) для ее описания. Основное назначение модели данных состоит в том, чтобы представить информационную картину в целом, без отвлекающих деталей, связанных с особенностями хранения.

Выделяют иерархическую, сетевую и реляционную модели данных. Эта последовательность отражает прогресс в понимании проблем обработки данных.

Наиболее абстрактной моделью является *реляционная модель данных*. Абстрактна она в том смысле, что в значительной степени ориентирована на интересы пользователя (программиста) и совершенно не несет в себе черт реального отображения на физическую память. Эта модель исторически возникла позже других. В ней все связи между объектами задаются путем явной фиксации идентификаторов объектов в записях. На первый взгляд, реляционная модель может быть представлена в виде однородных таблиц (отношений), которые напоминают стандартные последовательные файлы. Однако это очень упрощенная точка зрения. Дело в том, что последовательный файл предполагает определенную упорядоченность и обработку в соответствии с этой упорядоченностью, или, иными словами, предполагается вход со стороны одного поля (ключа сортировки).

Существенное отличие реляционной модели от обыкновенного последовательного файла заключается в том, что все столбцы в таблице предполагаются эквивалентными. Таким образом, реляционная модель предполагает очень высокую степень «интеллектуальности» систем управления базами данных.

#### 4.5.2. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

*Система управления базами данных* – составная часть автоматизированного банка данных, обеспечивающая работу прикладных программ с базой данных.

Одним из важнейших назначений СУБД является обеспечение независимости данных. Под этим термином понимается независимость данных и использующих их прикладных программ друг от друга в том смысле, что изменение одних не приводит к изменению дру-

гих. Система управления базами данных обеспечивает также защиту и секретность данных, восстановление БД после сбоев, ведение учета работы с БД. Однако это далеко не полный перечень того, что должна осуществлять СУБД для обеспечения интерфейса пользователей с БД и жизнеспособности всего автоматизированного банка данных.

Система управления данными имеет набор средств, которые обеспечивают определенные способы доступа к данным. Наиболее общими операциями, которые выполняются средствами СУБД, являются операции поиска, исправления, добавления и удаления данных. Операция поиска – главная среди перечисленных.

Степень реализации принципа независимости данных определяет гибкость СУБД. Учет особенностей обработки данных в какой-либо предметной области позволяет спроектировать специализированные СУБД, ориентированные на применение в АСУ предприятий. Существуют и универсальные СУБД, используемые для различных приложений. При настройке универсальных СУБД для конкретных приложений они должны обладать соответствующими средствами. К универсальным СУБД относятся, например, системы MS Access, MS SQL Server, MySQL, Oracle.

#### 4.5.3. ТЕХНОЛОГИЯ ИСПОЛЬЗОВАНИЯ СУБД

Рассмотрим технологию использования современных СУБД на примере СУБД Microsoft Access. Microsoft Access предоставляет технологию создания БД с помощью типовых инструментальных средств, рассчитанных на массового пользователя-непрограммиста. Несмотря на то, что Access ориентирована на конечного пользователя, в ней присутствует язык программирования и имеется возможность интеграции с другими программными средствами Microsoft Office.

Access (от англ. *access* – доступ) – популярная настольная система управления базой данных, рассчитанная на конечного пользователя. В то же время на небольшом предприятии (при объеме данных до 1 Гбайт) с количеством компьютеров не более 10 ресурсов Access вполне достаточно для обслуживания всего делопроизводства вместе со средствами Microsoft Office. Все пользователи могут обращаться к одной БД, установленной на одном компьютере, который может не быть сервером.

Access предоставляет средства защиты, с помощью которых решаются проблемы защищенного доступа к данным и их сохранности. Характерными чертами Access являются направленность на конечного пользователя (непрограммиста); сохранение общего подхода, принятого в построении продуктов Microsoft; массовость использования.

В Access для работы с данными служат процессор баз данных, средства быстрого построения интерфейса (Конструктор форм и отчетов), объекты доступа и манипулирования данными (таблицы, формы, запросы, макрокоманды, макросы, модули). Автоматизация рутинных типовых операций выполняется с помощью готовых визуальных средств или макрокоманд, объединяемых в макросы. Таким образом, пользователи Access могут обратиться к созданию процедур и функций для работы с данными. При этом, если не хватает возможностей визуальных готовых средств, обращаются к макрокомандам, а если и их возможностей недостаточно, можно использовать язык программирования. Он позволяет создавать свои массивы, типы данных, функции, приложения. Имеется возможность целиком создать БД с помощью программирования, когда в этом появится необходимость.

Технологии создания баз данных для ПК ориентированы на решение несложных задач с ограниченным объемом информации.

#### 4.6. МУЛЬТИМЕДИЙНЫЕ ТЕХНОЛОГИИ

Пакеты программ мультимедиа предназначены для использования в ПК для отображения и обработки аудио- и видеoinформации. Программы мультимедиа для ПК появились благодаря значительному росту вычислительных возможностей ПК и большим

достижениям в области производства оптических дисков. Дело в том, что при представлении аналоговой информации в цифровом виде требуются огромные объемы памяти: несколько минут видеофильма занимают десятки мегабайт памяти. Естественно, что работа с таким большим файлом возможна лишь при наличии быстродействующего процессора. Кроме того, распространение таких мультимедиа-приложений невозможно на традиционных магнитных дискетах, для этого необходимо использовать оптические компакт-диски.

**Мультимедиа** – это интерактивная технология, обеспечивающая работу с неподвижными изображениями, видеоизображением, анимацией, текстом и звуковым рядом. Одним из первых инструментальных средств создания технологии мультимедиа явилась гипертекстовая технология, которая обеспечивает работу с текстовой информацией, изображением, звуком, речью.

**Анимация** (от англ. *animate* – оживлять) – процесс создания движущихся графических изображений на экране дисплея; используется при проектировании различных объектов, моделировании физических явлений, в обучающих системах и игровых программах. В настоящее время разработаны специальные анимационные программные комплексы (пакеты), позволяющие любой объемный объект двигать и вращать в разных направлениях с разными скоростями. С помощью таких пакетов можно создавать мультипликационные фильмы.

Появление систем мультимедиа произвело революцию в образовании, компьютерном тренинге, бизнесе и др. Технология мультимедиа создала предпосылки для удовлетворения растущих потребностей общества. Она позволила заменить техноцентрический подход к развитию сфер экономики (планирование индустрии зависит от прогноза возможных технологий) на антропоцентрический (индустрия управляется рынком). Это дает возможность динамически отслеживать индивидуальные запросы мирового рынка, что отражается в тенденции перехода к мелкосерийному производству. Феномен мультимедиа демократизирует научное, художественное и промышленное творчество. Именно авторские технологии обеспечили процесс информатизации общества.

Самое широкое применение технология мультимедиа получила в сфере образования. Созданы видеэнциклопедии по многим школьным предметам, по музеям, городам, маршрутам путешествий, а также игровые ситуационные тренажеры, что сокращает время обучения. Формируется база знаний, в которой сконструированы живые миры; создается диалоговое кино, где потребитель может управлять ходом зрелища с клавиатуры дисплея, посредством реплик, если к компьютеру подключена плата распознавания речи.

Технология мультимедиа создает предпосылки для развития «домашней индустрии», что приводит к сокращению производственных площадей, увеличивает производительность труда. Особые перспективы она открывает для дистанционного обучения.

## **4.7. КОМПЬЮТЕРНЫЕ КОММУНИКАЦИИ**

### **4.7.1. ПЕРЕДАЧА ИНФОРМАЦИИ. ЛИНИИ СВЯЗИ, ИХ ОСНОВНЫЕ КОМПОНЕНТЫ И ХАРАКТЕРИСТИКИ**

Передача информации осуществляется различными способами. Особенно эффективен в смысле экономии времени способ дистанционной передачи данных по каналам связи. Для его осуществления необходимы специальные технические средства сбора и регистрации информации с датчиков, установленных на рабочих местах, и передачи ее в компьютер.

Дистанционно можно передавать как первичную информацию с мест ее возникновения, так и результатную – в обратном направлении. Результатная информация отражается на дисплеях, табло, печатающем устройстве. Поступление информации по

каналам связи в центр ее обработки в основном осуществляется двумя способами: на машинном носителе или непосредственно в компьютер с помощью специальных программных и аппаратных средств.

Дистанционная передача информации постоянно развивается и совершенствуется. Особое значение этот способ передачи информации имеет в многоуровневых межотраслевых системах, где его применение значительно ускоряет прохождение информации с одного уровня управления на другой и сокращает общее время обработки данных.

Появление компьютерных сетей, по сути, произвело своего рода техническую революцию, так как состоялось объединение технологии сбора, хранения, обработки и передачи информации на ЭВМ с техникой связи. Первые вычислительные сети (ВС) были разработаны в 1960-х гг. в США и СССР и объединяли большие ЭВМ и вычислительные центры (ВЦ). В США это была сеть ARPA, объединявшая 50 университетов и фирм, а в СССР – сеть Академии наук в Ленинграде.

Передача информации между территориально удаленными компонентами подобных распределенных систем осуществляется в основном с помощью стандартных телефонных и телеграфных каналов, а также витых пар проводов и коаксиальных кабелей связи. Современный прогресс в области оптоволоконной техники (использование световодов) позволяет резко повысить пропускную способность линий связи.

Беспроводные сетевые технологии, использующие частотные каналы передачи данных, представляют разумную альтернативу обычным проводным сетям и становятся все более привлекательными. Самое большое преимущество беспроводных технологий – это возможности, предоставляемые пользователям портативных компьютеров. Однако скорость передачи данных, достигаемая в беспроводных технологиях, не может пока сравниться с пропускной способностью кабеля, хотя она в последнее время и значительно выросла. Важно, что для перехода к беспроводной технологии не нужно менять уже имеющиеся сети.

В спутниковых технологиях физической средой передачи данных является эфир. Использование спутников оправданно в случае значительного удаления абонентов друг от друга при чрезмерном ослаблении посылаемых электромагнитных сигналов с большими посторонними шумами. Чтобы сигналы, направленные отправителем, не смешивались с сигналами к получателю, при работе со спутником прокладываются два частотных канала – для отправителя и получателя. Это позволяет избежать ошибок при передаче информации.

#### **4.7.2. КОМПЬЮТЕРНЫЕ ТЕЛЕКОММУНИКАЦИИ: НАЗНАЧЕНИЕ, СТРУКТУРА, РЕСУРСЫ**

Основным средством компьютерных телекоммуникаций являются вычислительные сети. Вычислительная сеть – это комплекс ЭВМ, вспомогательного оборудования, каналов связи и специального программного обеспечения для передачи данных между элементами сети. Под сетью обычно понимается структура организации технических средств, данных и программ.

Каналы связи – это технические устройства и физическая среда, обеспечивающие передачу данных. Они бывают аналоговые и цифровые, телефонные и телеграфные, радиочастотные и телевизионные, инфракрасные и оптические, а также выделенные и коммутируемые. Основная характеристика канала связи – его пропускная способность. Вычислительные сети позволяют автоматизировать управление производством, транспортом, материально-техническим снабжением в масштабе отдельных регионов и страны в целом.

Возможность концентрации в ВС больших объемов данных, общедоступность этих данных, а также программных и аппаратных средств обработки и высокая надежность их функционирования – все это позволяет улучшить информационное обслуживание пользователей и резко повысить эффективность применения вычислительной техники (ВТ).

Вычислительные сети предназначены для того, чтобы:

- организовывать параллельную обработку данных многими компьютерами;
- создавать распределенные БД, размещаемые в памяти отдельных ЭВМ;
- специализировать отдельные компьютеры для эффективного решения определенных классов задач;
- автоматизировать обмен информацией и программами между отдельными компьютерами и пользователями сети;
- резервировать вычислительные мощности и средства передачи данных с целью быстрого восстановления нормальной работы сети в случае выхода из строя некоторых из них;
- перераспределять вычислительные мощности между пользователями сети в зависимости от изменения их потребностей и сложности решаемых задач;
- стабилизировать и повышать уровень загрузки компьютеров и дорогостоящего периферийного оборудования;
- сочетать работу в широком диапазоне режимов: диалоговом, пакетном, в режимах «запрос – ответ», сбора, передачи и обмена информацией.

Как показывает практика, за счет расширения возможностей обработки данных, лучшей загрузки ресурсов и повышения надежности функционирования системы в целом стоимость обработки данных в ВС не менее чем в полтора раза ниже, по сравнению с обработкой аналогичных данных на автономных ЭВМ.

Вычислительные и информационные сети классифицируют по различным признакам. Сети, состоящие из программно-совместимых компьютеров, являются однородными, или гомогенными. Если компьютеры, входящие в сеть, программно несовместимы, то такая сеть является неоднородной, или гетерогенной.

*По типу организации передачи данных* различают сети с коммутацией каналов, с коммутацией сообщений, с коммутацией пакетов. Имеются сети, использующие смешанные системы передачи данных.

*По характеру реализуемых функций* сети подразделяют на:

- **вычислительные**, предназначенные для решения задач управления на основе вычислительной обработки исходной информации;
- **информационные**, предназначенные для получения справочных данных по запросу пользователей (чисто информационные сети используются редко);
- **смешанные**, в которых реализуются вычислительные и информационные функции (в настоящее время это основной вид ВС).

*По способу управления* вычислительные и информационные сети подразделяют на сети с децентрализованным, централизованным и смешанным управлением.

*По структуре построения* (топологии) сети подразделяют следующим образом: радиальные (звездообразные), кольцевые, многосвязные («каждый с каждым»), иерархическая, «общая шина» и др.

Основные функции систем передачи данных в условиях функционирования ВС заключаются в организации быстрой и надежной передачи информации произвольным абонентам сети, а также в сокращении затрат на передачу данных. Последнее особенно важно, так как за прошедшее десятилетие произошло увеличение доли затрат на передачу данных в общей структуре затрат на организацию сетевой обработки информации. Это объясняется главным образом тем, что затраты на техническое обеспечение ВС сократились за этот период примерно в десять раз, тогда как затраты на организацию и эксплуатацию каналов связи сократились только в два раза.

При использовании некоммутируемых каналов связи средства приема-передачи абонентских пунктов и ЭВМ постоянно соединены между собой, т. е. находятся в режиме on-line. В этом случае отсутствуют потери времени на коммутацию, обеспечиваются высокая степень готовности системы к передаче информации, более высокая надеж-

ность каналов связи и как следствие – достоверность передачи информации. Недостатками такого способа организации связи являются низкий коэффициент использования аппаратуры передачи данных и линий связи, высокие расходы на эксплуатацию сети. Рентабельность подобных сетей достигается только при условии полной загрузки этих каналов.

При коммутации абонентских пунктов и ЭВМ только на время передачи информации (т. е. нормальным режимом для них является режим off-line) принцип построения узла коммутации определяется способами организации прохождения информации в сетях передачи данных. Существуют три основных способа подготовки и передачи информации в сетях, основанных на коммутации: коммутация каналов, сообщений и пакетов.

**Коммутация каналов** заключается в установлении физического канала связи для передачи данных непосредственно между абонентами сети. При использовании коммутируемых каналов тракт (путь) передачи данных образуется из самих каналов связи и устройств коммутации, расположенных в узлах связи.

Установление соединения заключается в том, что абонент посылает в канал связи заданный набор символов, прохождение которых по сети через соответствующие узлы коммутации вызывает установку нужного соединения с вызываемым абонентом. Этот транзитный канал образуется в начале сеанса связи, остается фиксированным на период передачи всей информации и разрывается только после завершения передачи информации.

Коммутация каналов как способ соединения используется в основном в сетях, где требуется обеспечить непрерывность передачи сообщений (например, при использовании телефонных каналов связи и абонентского телеграфа). В этом случае связь абонентов возможна только при условии использования ими однотипной аппаратуры, одинаковых каналов связи, а также единых кодов.

При **коммутации сообщений** поступающая на узел связи информация передается в память узла связи, после чего анализируется адрес получателя. В зависимости от занятости требуемого канала сообщение либо передается в память соседнего узла, либо становится в очередь для последующей передачи. Таким образом, способ коммутации сообщений обеспечивает поэтапный характер передачи информации.

Способ коммутации сообщений обеспечивает независимость работы отдельных участков сети, что значительно повышает эффективность использования каналов связи при передаче одного и того же объема информации (в этом случае он может достигать 80–90 % от максимального значения). В системе с коммутацией сообщений происходит сглаживание несогласованности в пропускной способности каналов и более эффективно реализуется передача многоадресных сообщений (так как не требуется одновременного освобождения всех каналов между узлом-передатчиком и узлом-приемником). Передача информации может производиться в любое время, так как прямая связь абонентов друг с другом необязательна.

В последние годы появился еще один способ коммутации абонентов сети – так называемая **коммутация пакетов**. Этот способ сочетает в себе ряд преимуществ способов коммутации каналов и коммутации сообщений. При коммутации пакетов перед началом передачи сообщение разбивается на короткие пакеты фиксированной длины, которые затем передаются по сети. В пункте назначения эти пакеты вновь объединяются в первоначальное сообщение, а так как их длительное хранение в запоминающем устройстве узла связи не предполагается, пакеты передаются от узла к узлу с минимальной задержкой во времени. В этом отношении указанный способ близок к способу коммутации каналов.

При коммутации пакетов их фиксированная длина обеспечивает эффективность обработки пакетов, предотвращает блокировку линий связей и значительно уменьшает емкость требуемой промежуточной памяти узлов связи. Кроме того, сокращается время задержки при передаче информации, т. е. скорость передачи информации превышает аналогичную скорость при способе коммутации сообщений.

К недостаткам способа коммутации пакетов следует отнести односторонний характер связи между абонентами сети.

#### 4.7.3. ЛОКАЛЬНЫЕ КОМПЬЮТЕРНЫЕ СЕТИ

Прогресс в развитии микропроцессорной техники сделал ее доступной массовому потребителю, а высокая надежность, относительно низкая стоимость и простота общения с пользователем – не профессионалом в области вычислительной техники послужили основой для организации систем распределенной обработки данных, включающих от десятка до сотен ПК, объединенных в вычислительные сети. В отличие от вычислительных сетей, создаваемых на базе больших ЭВМ и охватывающих значительную территорию, сети на базе ПК получили название локальных, так как они ориентированы в первую очередь на объединение вычислительных машин и периферийных устройств, сосредоточенных на небольшом пространстве (например, в пределах одного помещения, здания, группы близлежащих зданий, расположенных на расстоянии нескольких километров друг от друга).

Появление локальных вычислительных сетей (ЛВС) позволило значительно повысить эффективность применения ВТ за счет более рационального использования аппаратных, программных и информационных ресурсов вычислительной системы, значительного улучшения эксплуатационных характеристик (в первую очередь повышения надежности) и создания максимальных удобств для работы конечных пользователей.

Сравнительно низкая стоимость, высокая живучесть, простота комплексирования и эксплуатации, оснащенность современными операционными системами различного назначения, высокоскоростными средствами передачи данных, оперативной и внешней памятью большой емкости – все это способствовало быстрому распространению и широкому применению ЛВС для автоматизации управленческой деятельности в учреждениях, на предприятиях, а также для создания на их основе информационных, измерительных и управляющих систем автоматизации технологических и производственных процессов.

Одной из главных проблем создания ЛВС является проблема аппаратной совместимости ВТ. В настоящее время вычислительные средства ЛВС в основном объединяются с помощью высоко- и низкоскоростных каналов передачи данных. Такие вычислительные сети получили название свободносвязанных, так как протекание вычислительных процессов в них может осуществляться асинхронно.

По топологическим признакам ЛВС подразделяют на сети:

- с общей шиной;
- кольцевые;
- иерархические;
- радиальные;
- многосвязные.

Под **методом доступа в ЛВС** понимается метод организации хранения и обмена данных в устройствах памяти, файлах, базах данных и сетях. Метод доступа **Ethernet** (от англ. *ether* – эфир и *net* – сеть), пользующийся наибольшей популярностью, обеспечивает высокую скорость передачи данных и надежность. Для Ethernet используется топология «общая шина». Поэтому сообщение, отправляемое одной рабочей станцией, принимается одновременно всеми остальными станциями, подключенными к общей шине. Но поскольку сообщение включает адреса станций отправителя и адресата, то другие станции это сообщение игнорируют. Ethernet – метод множественного доступа. При нем перед началом передачи рабочая станция определяет, свободен канал или занят. Если канал свободен, то станция начинает передачу.

**В серверных ЛВС** реализованы **две модели взаимодействия** пользователей с рабочими станциями – «файл-сервер» и «клиент-сервер».

В модели «*файл-сервер*» сервер обеспечивает доступ к файлам базы данных для каждой рабочей станции, и на этом его работа заканчивается. Например, если используется база данных типа «файл-сервер» для получения сведений о налогоплательщиках, проживающих на какой-либо конкретной улице Москвы, по сети будет передана вся таблица по территориальному округу, и решать, какие записи в ней удовлетворяют запросу, а какие – нет, приходится самой рабочей станции. Таким образом, работа модели «файл-сервер» приводит к перегрузке сети.

Модель «*клиент-сервер*» лишена этих недостатков. В этом случае прикладная система делится на две – внешнюю (обращенную к пользователю), называемую клиентом, и внутреннюю (обслуживающую), называемую сервером. Сервером является машина, обладающая ресурсами и предоставляющая их, а клиентом – потенциальный потребитель этих ресурсов. Роль ресурсов могут играть файловая система (файловый сервер), процессор (вычислительный сервер), база данных (сервер базы данных), принтер (принтер-сервер) и др. Так как сервер (или серверы) обслуживает одновременно многих клиентов, то на серверном компьютере должна функционировать многозадачная операционная система.

В модели «клиент-сервер» сервер играет активную роль, ибо программное обеспечение «заставляет» его «сначала подумать, а потом сделать». Потоки информации, текущие по сети, становятся меньшими, поскольку сервер сначала обрабатывает запросы, а затем посылает клиенту то, в чем он нуждается. Сервер также контролирует допустимость обращения к записям на индивидуальной основе, что обеспечивает большую безопасность данных.

Модель «клиент-сервер», созданная на основе ПК, характеризуется следующим:

- сеть содержит значительное количество серверов и клиентов;
- основу вычислительной системы составляют рабочие станции, каждая из которых функционирует как клиент и запрашивает информацию, которая находится на сервере;
- пользователь системы освобожден от необходимости знать, где находится требуемая ему информация: он просто запрашивает то, что ему нужно;
- система реализуется в виде открытой архитектуры, объединяющей ЭВМ различных классов и типов с различными системами.

#### 4.7.4. ГЛОБАЛЬНЫЕ КОМПЬЮТЕРНЫЕ СЕТИ

Сетевые технологии в настоящее время чрезвычайно разнообразны. К классификации вычислительных и информационных сетей, рассмотренной ранее, добавим классификацию по такому ключевому признаку, как охват территории. По охвату территории сети подразделяют на локальные, региональные (территориальные), федеральные и глобальные.

В 1997 г. в мире было зарегистрировано более 200 глобальных сетей, 56 из которых созданы в США, 16 – в Японии. Одна из первых глобальных сетей (ARPANet) охватывала всю территорию США, часть Европы и Азии. Сеть ARPANet доказала техническую возможность и экономическую целесообразность разработки крупных сетей для более эффективного использования компьютера и программного обеспечения. В Европе сначала были разработаны и внедрены международные сети, затем национальные. В 1972 г. в Австрии была создана сеть МИПСА, в 1979 г. к ней подсоединились 17 стран Европы, СССР, США, Канада и Япония. Сеть была создана для проведения фундаментальных исследовательских работ по проблемам энергетики, продовольствия, сельского хозяйства, здравоохранения и др. Кроме того, в рамках МИПСА была внедрена технология, позволяющая всем национальным институтам развивать связь друг с другом.

К созданной в 1960-х гг. в Ленинградском отделении Академии наук СССР сети в 1985 г. подключилась региональная сеть «Северо-Запад» с академическими центрами

Москвы и Риги. В 1980 г. была сдана в эксплуатацию система телеобработки статистической информации (СТОСИ), обслуживающая Главный ВЦ ЦСУ СССР в Москве и республиканские ВЦ в союзных республиках.

В течение последнего десятилетия получают все более широкое развитие глобальные вычислительные и информационные сети – уникальный симбиоз компьютеров и коммуникаций. Идет активное включение государств во всемирные сетевые структуры. Мировой системой компьютерных коммуникаций ежедневно пользуются более 30 млн человек. Возрастает потребность в средствах структурирования, накопления, хранения, поиска и передачи информации. Удовлетворению этих потребностей служат информационные сети и их ресурсы. Совместное использование ресурсов сетей (библиотек программ, баз данных, вычислительных мощностей) обеспечивается технологическим комплексом и средствами доступа.

Глобальные сети (Wide Area Network, WAN) – это телекоммуникационные структуры, объединяющие локальные информационные сети, имеющие общие протокол связи, методы подключения и протоколы обмена данными. Каждая из глобальных сетей (Internet, Bitnet, DECnet и др.) организовывалась для определенных целей, а в дальнейшем расширялась за счет подключения локальных сетей, использующих ее услуги и ресурсы.

Крупнейшей глобальной информационной сетью является Internet. В настоящее время в русскоязычной литературе закрепились наименование Internet и его синоним – Сеть, которые пишутся с прописной буквы.

Процесс совершенствования Internet идет непрерывно, большинство новаций незаметны для пользователей. Любой желающий может получить доступ к Сети. В России подключение к Internet началось в начале 1990-х гг.

#### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. По каким признакам классифицируют информационные технологии?
2. В чем отличие новых (современных) ИТ от традиционных?
3. Каковы назначение и основные функции текстового редактора?
4. Что вы знаете о графических редакторах?
5. Что такое электронная таблица?
6. Как вы представляете базу данных?
7. Какие способы организации баз данных вы знаете?
8. Раскройте понятие СУБД.
9. Дайте определение понятия «мультимедиа».
10. Расскажите о технических средствах мультимедийных технологий.
11. Каково назначение вычислительных сетей?
12. Дайте классификацию ВС.
13. Что такое коммутация каналов?
14. Перечислите преимущества и недостатки коммутации сообщений и пакетов.
15. Расскажите об электронной почте.
16. Какие признаки положены в основу классификации ЛВС?
17. Каким образом организуется обмен информацией в ЛВС?
18. Перечислите сходства и различия между локальными и глобальными вычислительными сетями.
19. Дайте характеристику методам доступа в ЛВС.
20. В чем состоит отличие моделей «файл-сервер» и «клиент-сервер»?
21. Что такое компьютерный вирус?
22. Дайте характеристику основных ресурсов Internet.

## 5. БАЗЫ ДАННЫХ

### 5.1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

#### 5.1.1. УРОВНИ ДАННЫХ БД

В базе данных отражается информация об определенной предметной области. **Предметной областью** называется часть реального мира, представляющая интерес для данного использования. В автоматизированных информационных системах отражение предметной области представлено моделями данных нескольких уровней (рис. 5.1). Число уровней моделей зависит от особенностей СУБД.

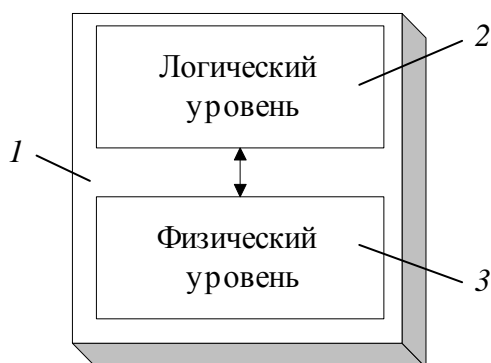


Рис. 5.1. Уровни моделей данных:  
1 - база данных; 2 - даталогическая модель данных; 3 - физическая модель данных

**Даталогическая модель** базы данных является моделью логического уровня и представляет собой отображение логических связей между элементами данных безотносительно к их содержанию и среде хранения. Эта модель строится в терминах информационных единиц, допустимых в конкретной СУБД. Этап создания даталогической модели называется **даталогическим проектированием**.

Для привязки даталогической модели к среде хранения используется модель данных физического уровня – **физическая модель данных**. Эта модель определяет используемые запоминающие устройства, способы физической организации данных в среде хранения. Модель физического уровня также строится с учетом возможностей, предоставляемых СУБД. Соответствующий этап проектирования БД называется **физическим проектированием**.

В настоящее время наблюдается тенденция к сокращению работ на стадии физического проектирования или скрытию их от проектировщика.

Для того чтобы спроектировать структуру базы данных, необходима исходная информация о предметной области. Желательно, чтобы эта информация была представлена в формализованном виде. Информация, требуемая для проектирования БД, мало зависит от особенностей СУБД.

Описание предметной области, выполненное без ориентации на используемые в дальнейшем программные и технические средства, называется **инфологической моделью предметной области (ИЛМ)**.

#### 5.1.2. ЭТАПЫ ПРОЕКТИРОВАНИЯ БД

Инфологическая модель предметной области строится первой (рис. 5.2). Затем на ее основе строится даталогическая модель. После этого может строиться физическая модель. При проектировании БД возможен возврат на предыдущие уровни. При этом возможны два вида возвратов:

- первый тип обусловлен необходимостью пересмотра результата проектирования (например, для улучшения полученных характеристик, «обхода» ограничений и т.п.);
- второй тип вызван необходимостью уточнения предыдущей модели (обычно инфологической) с целью получения дополнительной информации для проектирования или при выявлении противоречий в модели.

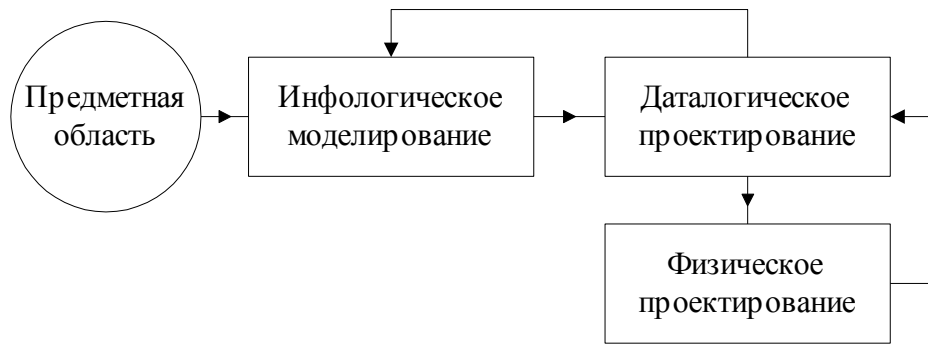


Рис. 5.2. Этапы проектирования БД

### 5.1.3. ИНФОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Существует несколько методик разработки инфологической модели. Рассмотрим метод ER-диаграмм. **ER-метод** (*entity-relationship* – «сущность-связь») разработан Питером Ченом в 1976 г.

#### Сущности и связи

Предположим, что проектируется база данных, предназначенная для хранения информации об авторах и о тех книгах, которые они написали. Двумя главными объектами, или сущностями, представляющими в данном случае интерес, являются **АВТОР** и **КНИГА**. Эти две сущности соотносятся с помощью связи **ПИШЕТ**, что позволяет нам сказать:

#### **АВТОР ПИШЕТ КНИГУ**

Связь **ПИШЕТ**, существующая между двумя сущностями **АВТОР** и **КНИГА**, может быть графически представлена несколькими способами. С помощью диаграммы **ER-экземпляров** (рис. 5.3, а) можно показать, какую книгу пишет автор. Диаграмма **ER-типа** (рис. 5.3, б) содержит такую же общую информацию, как и диаграммы **ER-экземпляров**. В ней каждый автор идентифицируется номером-автора (**на**), а каждая книга – номером-книги (**нк**). Эти диаграммы содержат всю информацию, необходимую для генерации таблиц в начале проектирования БД.

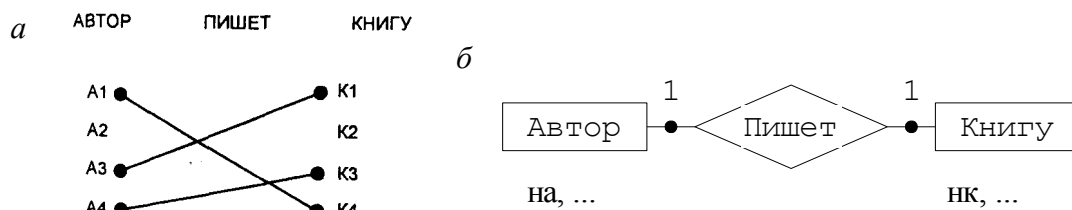


Рис. 5.3. Диаграммы: а - ER-экземпляров; б - ER-типа

**Сущность** определяется как некоторый объект, представляющий интерес. Этот объект должен иметь экземпляры, отличающиеся друг от друга и допускающие однозначную идентификацию. Единственный определяющий признак, который может помочь в нахождении сущностей, состоит в том, что сущность – это, как правило, существительное. Примерами сущностей могут служить машины, банковские счета, колледжи, служащие и т.п. В нашем примере (рис. 5.3) сущностями являются **КНИГА** и **АВТОР**, в то время как отдельные **экземпляры** каждой сущности идентифицируются с помощью **номера-книги** и **номера-автора** соответственно.

**Класс принадлежности сущности** характеризует обязательность включения каждого экземпляра сущности в связь.

**Связь** представляет собой соединение между двумя или более экземплярами сущностей. При поиске связей в основном следует полагаться на то обстоятельство, что связь обычно выражается глаголом. Типичными примерами связей между двумя сущностями являются:

- служащие **РАБОТАЮТ** в отделах;
- студенты **ИЗУЧАЮТ** предметы;
- рабочие **ОБСЛУЖИВАЮТ** механизмы.

**Атрибут** есть свойство сущности. Например, атрибутами, которые могут быть свойствами сущности КНИГА, являются: номер книги, количество страниц, тираж, цена и т.д.

Особенность ER-метода заключается в том, что два проектировщика могут рассматривать одну и ту же проблему с разных точек зрения, получая в итоге различные наборы сущностей и связей. Определение лучшей из нескольких возможных наборов является вопросом личного предпочтения.

В некоторых случаях для идентификации каждого экземпляра сущности может понадобиться набор атрибутов. Атрибут, или набор атрибутов, используемый для идентификации экземпляра сущности, называется **ключом сущности**. Каждый экземпляр связи однозначно определяется набором ключей сущностей, соединяемых этой связью. Таким образом, <A1, K3> является **ключом связи**. На данном этапе процесса проектирования единственными необходимыми атрибутами являются те, которые требуются для формирования ключей сущностей. Другие атрибуты вместе с определенными над ними функциональными зависимостями будут добавлены на более поздних этапах проектирования.

На диаграммах ER-типа сущности представляются в виде прямоугольников, а связи – в виде ромбов. Ниже каждой сущности размещается атрибут, или набор атрибутов, являющийся ключом для данной сущности.

В большинстве случаев для определения набора отношений проектируемой базы данных используются диаграммы ER-типа (ER-диаграммы), а не диаграммы ER-экземпляров.

#### 5.1.4. Степень связи

Важной характеристикой связи между двумя (или несколькими) сущностями является **степень связи**.

При степени связи, равной 1:1 (один-к-одному), для сущностей **АВТОР** и **КНИГА** могут существовать формы диаграммы ER-экземпляров (рис. 5.4) для следующих случаев:

а) каждый автор пишет не более одной книги, и каждая книга пишется не более чем одним автором, т. е. допускается наличие авторов, не пишущих ни одну книгу, а также книг, которые не пишет ни один из авторов. Таким образом, ни один автор не должен писать более одной книги, и ни одна книга не должна быть написана более чем одним автором;

б) каждый автор пишет только одну книгу, и каждую книгу пишет не более чем один автор. Здесь допускаются книги, которые не пишет никто;

в) каждый автор пишет не более одной книги, и каждую книгу пишет только один автор. Здесь допускается наличие авторов, не пишущих ни одной книги;

г) каждый автор пишет только одну книгу, и каждую книгу пишет только один автор, т. е. не допускается наличие авторов, не пишущих ни одной книги, а также книг, которые не пишет никто.

Тот факт, что каждый экземпляр сущности, расположенный как в левой, так и в правой частях диаграммы, связывается не более чем с одним экземпляром сущности, расположенным в противоположной части диаграммы, дает основание определить каждую из диаграмм экземпляров (см. рис. 5.4), как имеющую степень связи 1:1.

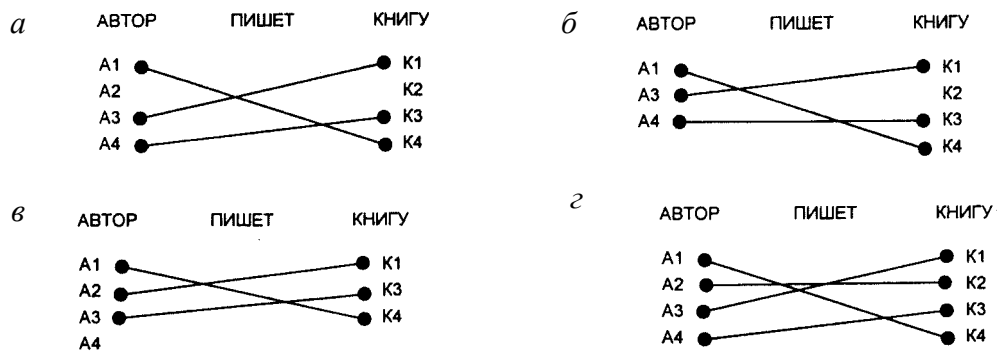


Рис. 5.4. Возможные формы диаграммы ER-экземпляров, имеющей степень связи 1:1:

- а - класс принадлежности ни одной сущности не является обязательным;
- б - класс принадлежности сущности АВТОР является обязательным;
- в - класс принадлежности сущности КНИГА является обязательным;
- г - класс принадлежности обеих сущностей является обязательным

Более компактная форма представления информации (см. рис. 5.4) проиллюстрирована на рисунке 5.5.

Различия ER-диаграмм (рис. 5.5), соответствующих диаграммам ER-экземпляров (см. рис. 5.4), являются следствием того, должны или не должны все экземпляры сущности участвовать в связи. Для случая (рис. 5.5, а) не выставляется требование участия в связи всех экземпляров обеих сущностей. Для случая (рис. 5.5, б) все экземпляры сущности АВТОР обязательно должны участвовать в связи, а экземпляры сущности КНИГА – не обязательно. Для случая (рис. 5.5, в) требуется участие в связи каждого экземпляра сущности КНИГА и допускается неучастие некоторых экземпляров сущности АВТОР. Для случая (рис. 5.5, г) требование обязательного участия в связи накладывается на все экземпляры обеих сущностей.

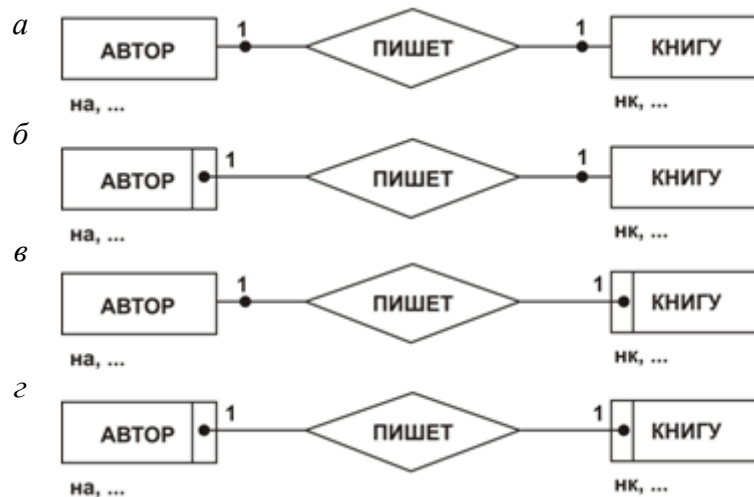


Рис. 5.5. Диаграммы ER-типов, соответствующие диаграммам ER-экземпляров (см рис. 5.4)

Если экземпляры данной сущности должны участвовать в связи, то участие называется **обязательным**, и этот факт отмечается маленьким черным кружком, помещенным в блок, смежный с блоком сущности. Если экземпляры данной сущности могут не участвовать в связи, то участие называется **необязательным**, и кружок располагается вне блока сущности. Класс принадлежности сущности должен быть либо обязатель-

ным, либо необязательным, и определяться правилами, регламентирующими деятельность организации. Единицы в обеих частях связей (см. рис. 5.5), говорят о том, что степени всех связей относятся к типу 1:1. В диаграммах ER-типа непосредственно под блоком каждой сущности выписывается и выделяется подчеркиванием ключ этой сущности: *на* (номер-автора) для сущности *АВТОР* и *нк* (номер-книги) для сущности *КНИГА*. Многоточие после каждого из этих атрибутов указывает на то, что никакие другие возможно имеющиеся атрибуты соответствующей сущности не могут быть частью ее ключа. Эти атрибуты будут добавлены после разработки таблиц.

Ни одно из правил, регламентирующих работу организации и используемых при составлении рассмотренных диаграмм ( см. рис. 5.3 – 5.5), не допускает, что автор пишет более одной книги, а также что одну книгу пишет более одного автора. Очевидно, что это в достаточной степени надуманные ограничения, существуют есть и другие варианты:

- 1) каждый автор может писать одновременно несколько книг, но каждая книга пишется не более чем одним автором;
- 2) каждый автор пишет не более одной книги, но каждую книгу может писать сразу несколько авторов;
- 3) каждый автор может писать несколько книг, и каждую книгу может писать несколько авторов.

Каждый из этих вариантов имеет несколько подвариантов, а именно: класс принадлежности может быть обязательным или необязательным – для одной из двух, ни для одной или для обеих сущностей.

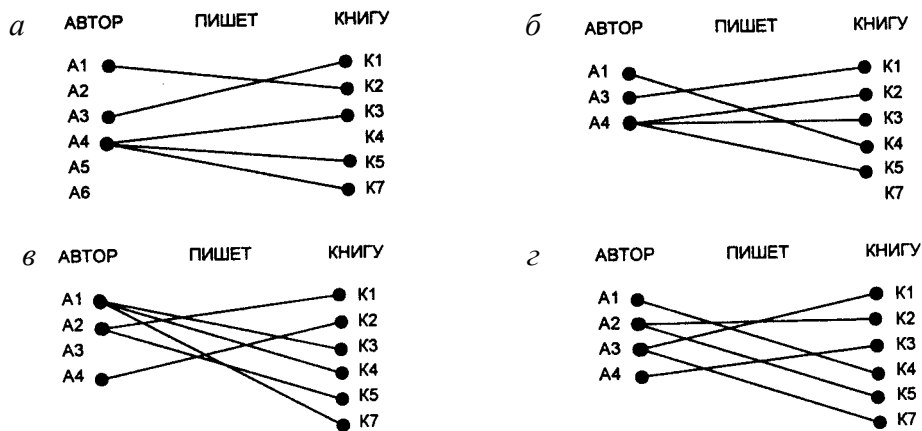


Рис. 5.6. Примеры диаграмм ER-экземпляров, имеющих степень связи 1:n:  
 а - класс принадлежности ни одной сущности не является обязательным;  
 б - класс принадлежности сущности АВТОР является обязательным;  
 в - класс принадлежности сущности КНИГА является обязательным;  
 г - класс принадлежности обеих сущностей является обязательным

Рассмотрим диаграммы ER-экземпляров для первого варианта, когда каждый автор может писать одновременно несколько книг, но каждая книга пишется не более чем одним автором (рис. 5.6) и эквивалентные им ER-диаграммы (рис. 5.7).

В примерах (рис. 5.7, а) ни один из классов принадлежности не является обязательным. В примерах (рис. 5.7, б и в) класс принадлежности является обязательным для одной из двух сущностей. В примерах (рис. 5.7, г) класс принадлежности является обязательным для обеих сущностей. В каждом из этих примеров степень связи имеет тип 1:n (*один-ко-многим*), вследствие того, что каждый экземпляр сущности *КНИГА* может

быть связан не более чем с одним экземпляром сущности *АВТОР* (отсюда получаем 1), а каждый экземпляр автора может быть связан более чем с одним экземпляром книги (отсюда получаем n).

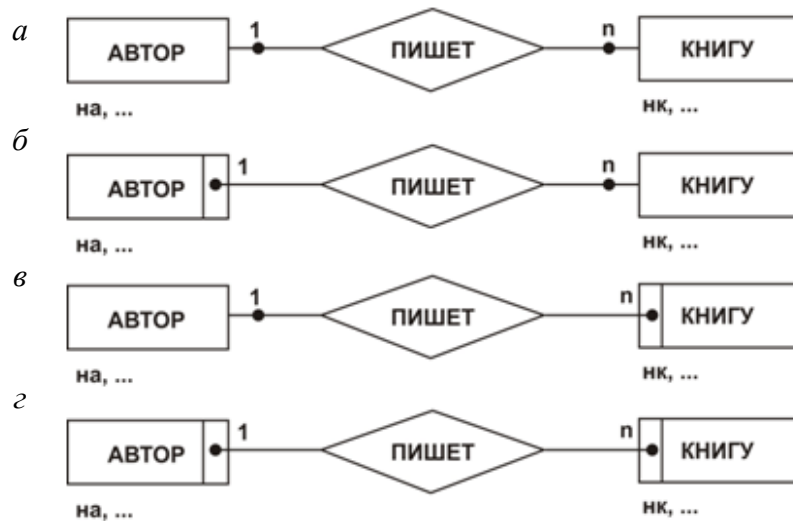


Рис. 5.7. ER-диаграммы, соответствующие диаграммам ER-экземпляров (см. рис. 5.6)

Теперь рассмотрим различные диаграммы ER-экземпляров для второго варианта (рис. 5.8), когда каждый автор пишет не более одной книги, но каждую книгу может писать сразу несколько авторов, и эквивалентные им ER-диаграммы (рис. 5.9).

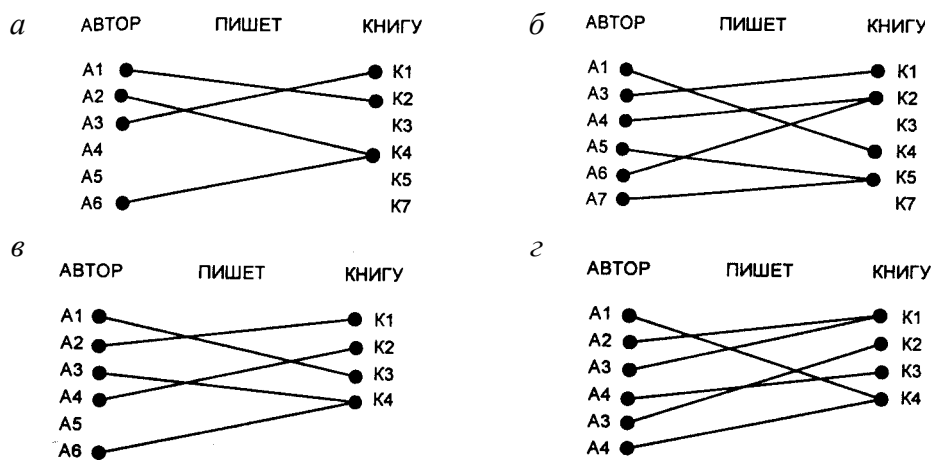


Рис. 5.8. Примеры диаграмм ER-экземпляров, имеющих степень связи  $n:1$ :  
 а - класс принадлежности ни одной сущности не является обязательным;  
 б - класс принадлежности сущности *АВТОР* является обязательным;  
 в - класс принадлежности сущности *КНИГА* является обязательным;  
 г - класс принадлежности обеих сущностей является обязательным

В примерах (рис. 5.8, а) класс принадлежности не является обязательным для сущностей как левой, так и правой частей. В примерах (рис. 5.8, б и в) класс принадлежности является обязательным для сущности либо левой, либо правой частей. В примерах (рис. 5.8, г) класс принадлежности является обязательным для обеих сущностей. В каждом из этих примеров степень связи имеет тип  $n:1$  (*многие-к-одному*), так как экземпляр сущности *КНИГА* может быть связан с более чем одним экземпляром

сущности **АВТОР** ( $n$ ), а каждый экземпляр сущности АВТОР связывается с не более чем одним экземпляром сущности **КНИГА** (1).

Наконец, рассмотрим диаграммы ER-экземпляров для третьего варианта (рис. 5.10), когда каждый автор пишет только одну книгу, и каждую книгу пишет только один автор, т.е. не допускается наличие авторов, не пишущих ни одной книги, а также книг, которые не пишет никто. Эквивалентные им ER-диаграммы показаны на рисунке 5.11.

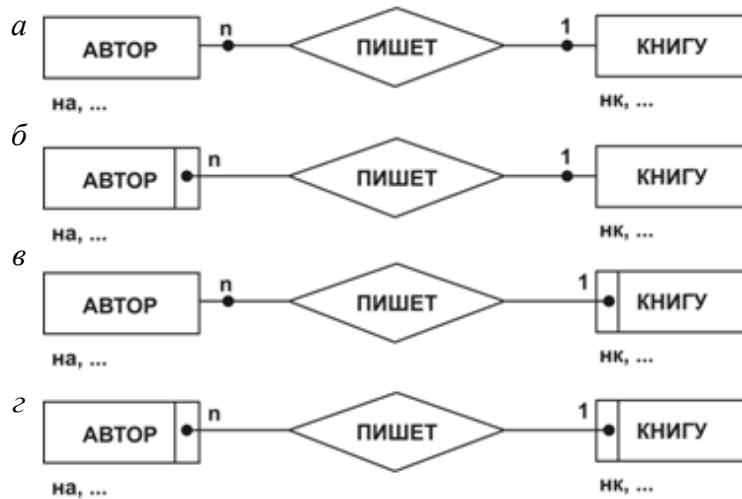


Рис. 5.9. ER-диаграммы, соответствующие диаграммам ER-экземпляров (см. рис. 5.8)

В примерах (рис. 5.10, а) класс принадлежности не является обязательным ни для одной из сущностей. В примерах (рис. 5.10, б и в) класс принадлежности является обязательным для одной из сущностей. В примерах (рис. 5.10, г) класс принадлежности является обязательным для обеих сущностей.

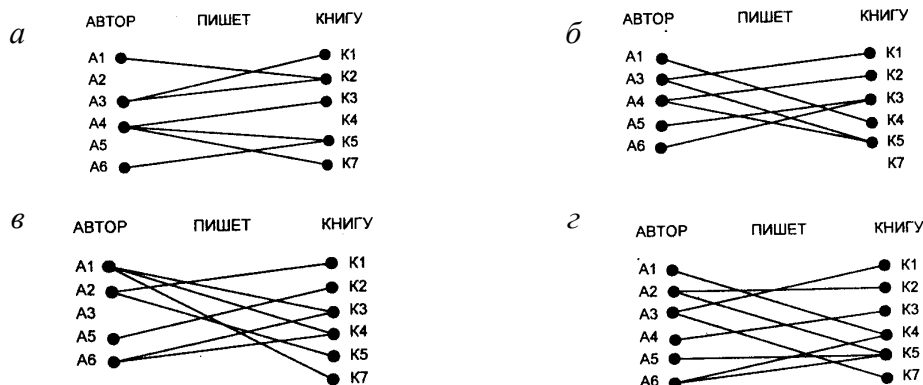


Рис. 5.10. Примеры диаграмм ER-экземпляров, имеющих степень связи  $m:n$ :  
 а - класс принадлежности ни одной сущности не является обязательным;  
 б - класс принадлежности сущности **АВТОР** является обязательным;  
 в - класс принадлежности сущности **КНИГА** является обязательным;  
 г - класс принадлежности обеих сущностей является обязательным

В каждом из этих примеров степень связи имеет тип  $m:n$  (**многие-ко-многим**), так как каждый экземпляр сущности **КНИГА** может быть связан с более чем одним экземпляром сущности **АВТОР** ( $m$ ), и каждый экземпляр сущности АВТОР может быть связан с более чем одним экземпляром сущности **КНИГА** ( $n$ ).

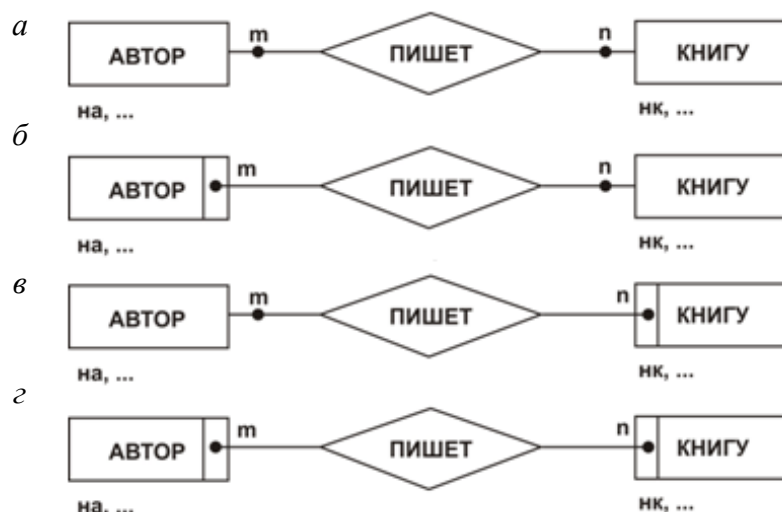


Рис. 5.11. ER-диаграммы, соответствующие диаграммам ER-экземпляров (см. рис. 5.10)

После построения ER-диаграмм парных сущностей составляют общую ER-диаграмму рассматриваемой предметной области (рис. 5.12).

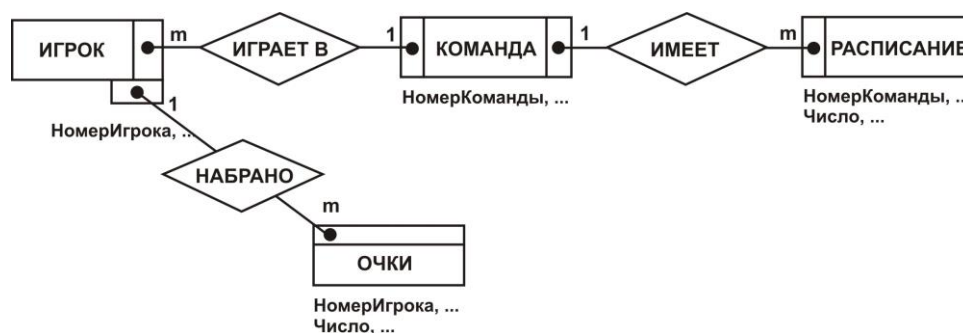


Рис. 5.12. ER-диаграмма любителя футбольной статистики

## 5.2. СОЗДАНИЕ БАЗЫ ДАННЫХ, СОСТОЯЩЕЙ ИЗ ОДНОЙ ТАБЛИЦЫ

### Цели работы:

- 1) познакомиться с основными понятиями базы данных;
- 2) научиться создавать таблицу базы данных в режиме Конструктор;
- 3) освоить переход из режима Конструктор в режим таблицы;
- 4) освоить основные приемы заполнения и редактирования таблиц базы данных;
- 5) познакомиться с простой сортировкой значений таблицы;
- 6) познакомиться с поиском записей по образцу;
- 7) научиться сохранять и загружать базу данных;
- 8) научиться распечатывать таблицы.

### Этапы работы:

- 1) **проектирование структуры базы данных.** Предположим, что необходимо создать самый простой вариант базы данных, когда вся информация хранится в одной таблице;
- 2) **конструирование структуры будущих таблиц базы данных.** Таблицу будем создавать в режиме «Конструктор». В ней будет 7 полей (код, фамилия, имя, отчество, год рождения, группа, факультет) и 10 записей. Поле «код» – уникальный ключ записи;

3) **создание схемы базы данных.** При наличии одной таблицы схема базы данных простая: состоит из одной этой таблицы, т.е. ее можно специально не создавать;

4) **ввод данных в таблицы.** В данном варианте будет только одна таблица. Таблицу будем создавать в режиме «**Конструктор**», а заполнять – **в режиме таблицы**.

### Задание 1

#### Создание новой базы данных

Назначение любой системы управления базами данных – создание, ведение и обработка баз данных. Как в текстовом редакторе можно подготовить много разных документов, так в СУБД Access можно создать много разных баз данных.

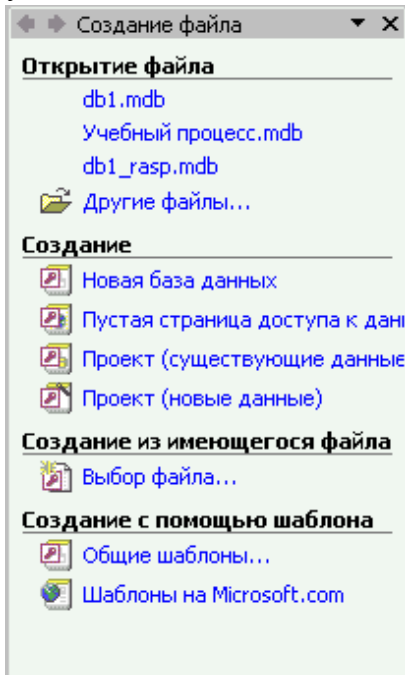


Рис. 5.13. Создание файла

Система управления базами данных предоставляет значительные возможности по работе с хранящимися данными, их обработке и совместному использованию. Можно выбирать любые поля, форматы полей, сортировать данные, вычислять итоговые значения. Можно отбирать интересующие вас данные по какому-либо признаку, менять их, удалять, копировать в другие таблицы.

Можно производить обмен данными между компонентами СУБД Access и другими приложениями. Это могут быть рисунки, диаграммы и т.д. Поддерживается экспорт и импорт данных из текстовых файлов и электронных таблиц.

При коллективном использовании СУБД Access дает возможность защитить информацию так, что разные пользователи имеют разные права по просмотру или изменению информации: при этом предусмотрены средства обеспечения целостности данных.

Каждая база данных хранится на диске в виде файла с расширением **mdb**.

Для создания новой базы данных выполните следующие действия:

1. Запустите программу **Access**. Перед вами откроется окно «**Access**», в котором появится меню, представленное на рис. 5.13.

2. Выберите команду «**Новая база данных**». Появится диалоговое окно (рис. 5.14). В поле «**Имя файла**» в качестве имени базы данных введите «**Студент**».

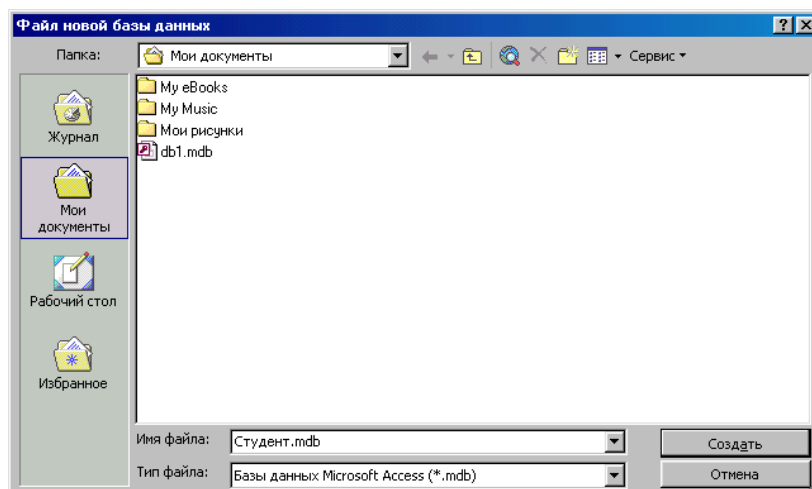


Рис. 5.14. Файл новой базы данных

3. Щелкните по кнопке «**Создать**» (см. рис. 5.14). Появится главное окно базы данных (рис. 5.15), в котором представлены все объекты базы данных: «**Таблицы**», «**Формы**», «**Запросы**», «**Отчеты**» и др.

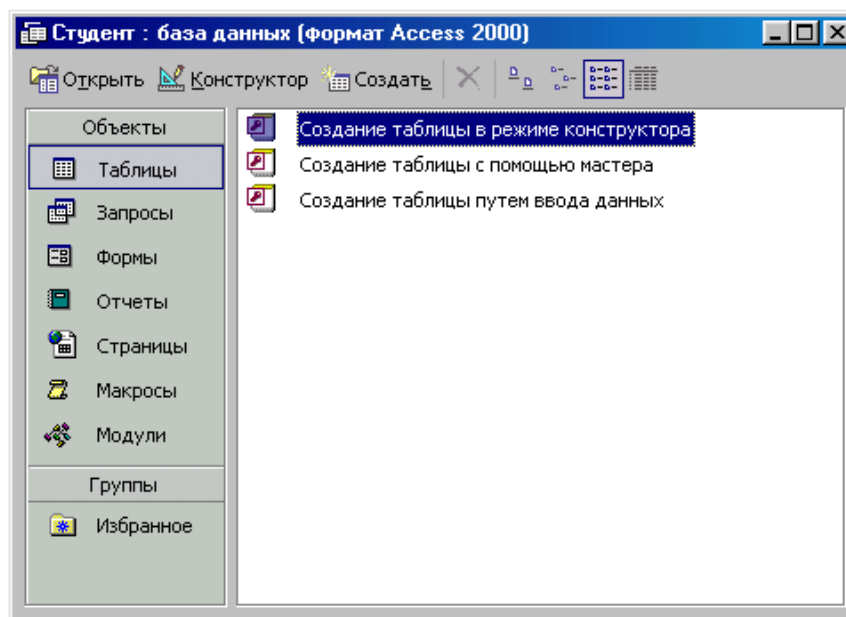


Рис. 5.15. Создание таблицы

## Задание 2

### Создание таблицы с информацией о студентах

Основная информация хранится в таблицах. **Таблица** - совокупность записей. Столбцы в таблице называются **полями**, а строки – **записями**. Количество записей в таблице ограничивается емкостью жесткого диска. Допустимое количество полей – 255. Таблиц в базе данных может быть несколько. Сведения по разным вопросам следует хранить в разных таблицах. Для работы таблицу необходимо открыть. Перед окончанием работы ее следует закрыть, предварительно сохранив все изменения, произведенные в ходе работы.

С таблицами можно работать в двух режимах – **таблицы** и **конструктора**. Переход из режима таблицы в режим конструктора таблицы и обратно производится щелчком по кнопке «**Вид**», расположенной на панели инструментов. **Ключевое поле** – поле с уникальными записями. Таблицы связываются (дается указание на соответствие записей) по ключам; ключ может состоять из одного или из нескольких полей.

Все объекты базы данных можно импортировать, т.е. копировать из других баз данных, а не вводить заново. Если таблицы были связаны в старой базе данных, то они таким же образом будут связаны и в новой.

В режиме таблицы обычно просматривают, добавляют и изменяют данные. Можно также добавлять или удалять столбцы таблицы, изменять внешний вид таблицы (ширину столбцов, их порядок, вид и цвет шрифта и т.д.). Можно проверить орфогра-

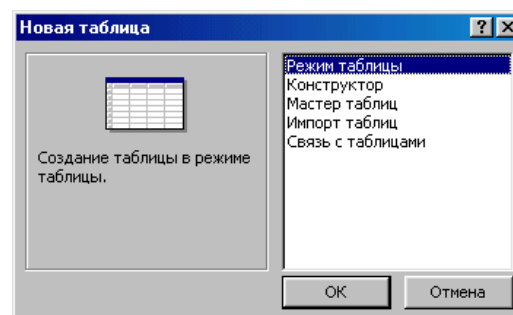


Рис. 5.16. Создание таблицы в режиме таблицы

фию и напечатать табличные данные, фильтровать и сортировать записи. В режиме конструктора таблицы можно создать новую таблицу или изменить поля старой.

Для создания таблицы с информацией о студентах выполните следующие действия:

В главном окне выберите тип создаваемого документа (выберите соответствующую закладку). Вы создаете таблицу, поэтому выберите закладку «**Таблица**». Щелкните по кнопке «**Создать**».

Появится диалоговое окно «**Новая таблица**» (рис. 5.16), в нем представлено несколько вариантов создания таблицы. Выберите «**Конструктор**» и щелкните по кнопке «**ОК**». Появится окно «**Конструктора**» (рис. 5.17).

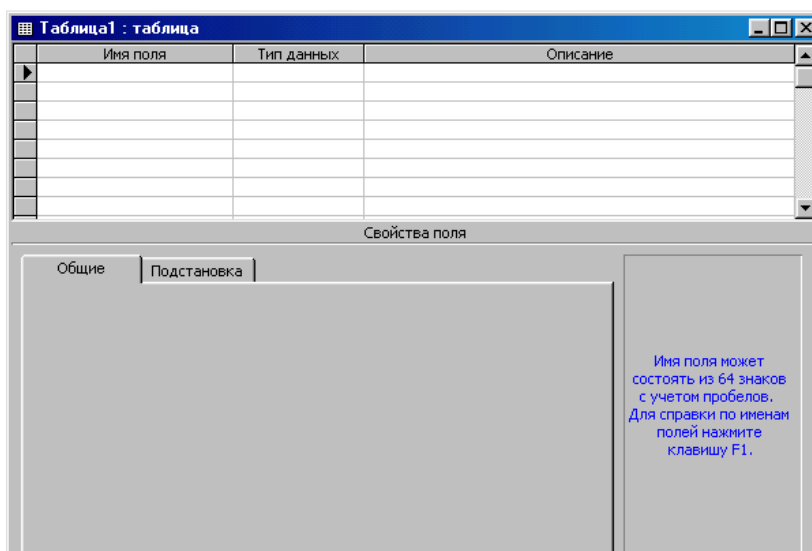


Рис. 5.17. Новая таблица

В левой верхней ячейке таблицы (рис. 5.18) введите имя поля «**Фамилия**». В соседней ячейке появится тип данных, по умолчанию он задается «**Текстовый**». Любой другой тип данных выбирается с помощью ниспадающего меню.

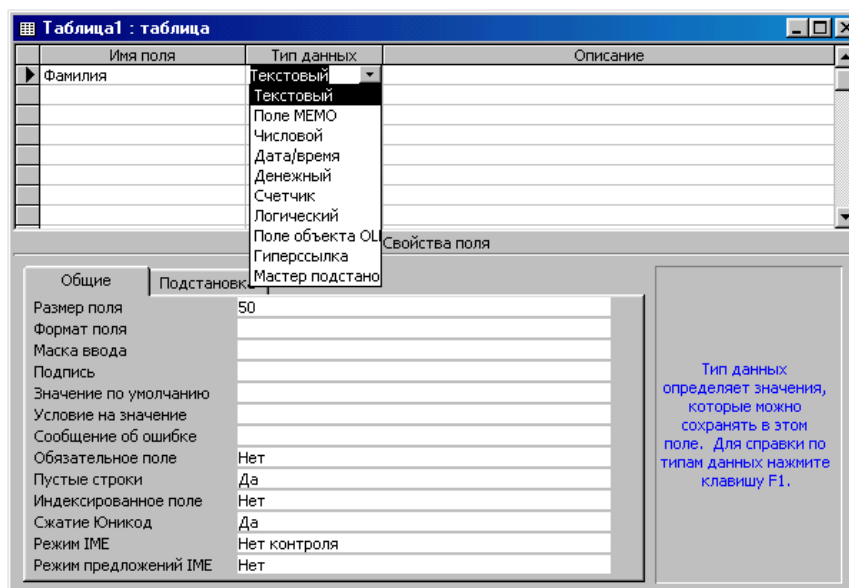


Рис. 5.18. Заполнение таблицы

Заполните поля в «**Конструкторе**» данными из табл. 5.1. Общие свойства поля, выводимые в нижней части «**Конструктора**», оставляем по умолчанию.

Таблица 5.1

Данные для заполнения полей в «Конструкторе»

Имя поля	Тип данных
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Год рождения	Числовой
Группа	Текстовый
Факультет	Текстовый

Сохраните таблицу, выбрав меню «**Файл**> **Сохранить**».

В появившемся окне наберите имя таблицы «**Список**» и щелкните по кнопке «**ОК**». Появится запрос (рис. 5.19) на создание ключевого поля – уникального поля записи, по которому удобно связывать таблицы. В данном варианте будут самостоятельно появляться числа – номера записей.

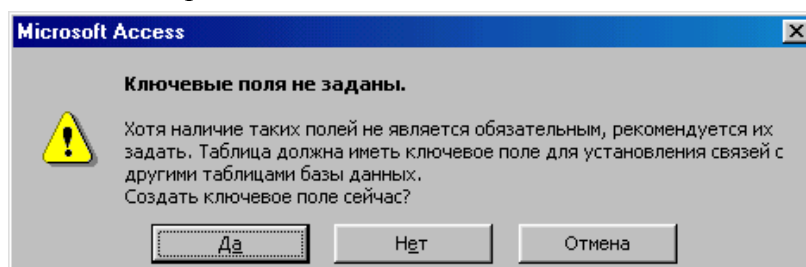



Рис. 5.19. Запрос создания ключевого поля

Выберите кнопку «**Да**».

Перейдите в режим таблицы, щелкнув по кнопке . Ввод данных вы будете производить в этом режиме, заполняя клетки таблицы. Значение поля «**Kod**» будет меняться автоматически. Если закончить ввод в ячейку нажатием на клавишу «**Enter**», то маркер перейдет в следующую ячейку.

Заполните базу данных значениями из табл. 5.2.

Таблица 5.2

Данные для заполнения базы данных

Код	Фамилия	Имя	Отчество	Год рождения	Группа	Факультет
1	Иванникова	Анна	Ивановна	1984	1ТМ	ИКПМТО
2	Баранова	Ирина	Алексеевна	1983	0ТС	ФАКС
3	Корнилова	Ольга	Владимировна	1984	1ТМ	ИКПМТО
4	Воробьев	Алексей	Петрович	1983	0ТС	ФАКС
5	Воробьев	Алексей	Иванович	1984	1МС	ИКПМТО
6	Воробьев	Олег	Григорьевич	1985	2ИС	ФКТ
7	Скоркин	Александр	Евгеньевич	1982	2ИС	ФКТ
8	Володина	Анна	Алексеевна	1984	1МС	ИКПМТО
9	Новоселов	Алексей	Антонович	1983	0ТС	ФАКС
10	Александрова	Елена	Алексеевна	1984	1ТМ	ИКПМТО

Сохраните введенные данные. В результате вы получили таблицу, с которой можно будет работать (рис. 5.20).

Код	Фамилия	Имя	Отчество	Год рождения	Группа	Факультет
1	Иванникова	Анна	Ивановна	1984	1ТМ	ИКПМТО
2	Баранова	Ирина	Алексеевна	1983	0ТС	ФАКС
3	Корнилова	Ольга	Владимировна	1984	1ТМ	ИКПМТО
4	Воробьев	Алексей	Петрович	1983	0ТС	ФАКС
5	Воробьев	Алексей	Иванович	1984	1МС	ИКПМТО
6	Воробьев	Олег	Григорьевич	1985	2ИС	ФКТ
7	Скоркин	Александр	Евгеньевич	1982	2ИС	ФКТ
8	Володина	Анна	Алексеевна	1984	1МС	ИКПМТО
9	Новоселов	Алексей	Антонович	1983	0ТС	ФАКС
10	Александрова	Елена	Алексеевна	1984	1ТМ	ИКПМТО
*	(Счетчик)			0		

Запись: 1 из 10

Рис. 5.20. Заполненная таблица с информацией о студентах

### Задание 3

#### Редактирование ячеек

Для редактирования ячеек выполните следующие действия:

1. Замените фамилию Иванникова на Иванова. Для этого выделите редактируемую ячейку и наберите новую фамилию.

Замените год рождения на 1983. Для этого следует щелкнуть мышкой по нужной ячейке, и она откроется для редактирования. Удалите цифру 4 и введите вместо нее цифру 3.

### Задание 4

#### Сортировка значений таблицы

Чтобы произвести сортировку значений, нужно поставить курсор в любую ячейку сортируемого столбца и щелкнуть по кнопке на панели инструментов:





– если нужно отсортировать по возрастанию;





– если нужно отсортировать по убыванию.


Отсортируйте:

1) **фамилии** – по алфавиту (поставьте маркер на любую фамилию в столбце с названием «**Фамилия**» и щелкните мышкой по кнопке 

2) **имя** – по алфавиту (поставьте маркер на любое имя в столбце с названием «**Имя**» и щелкните мышкой по кнопке 

3) **факультет** – по убыванию (поставьте маркер на любой факультет в столбце с названием «**Факультет**» и щелкните мышкой по кнопке 


4) **год рождения** – по убыванию (поставьте маркер на любой год рождения в столбце с названием «**Год рождения**» и щелкните мышкой по кнопке 

5) **группа** – по возрастанию (поставьте маркер на любую группу в столбце с названием «**Группа**» и щелкните мышкой по кнопке 

**Задание 5****Знакомство с функциями: сохранить, закрыть, открыть**

Выполните следующие действия:

1. Щелкнув по кнопке , сохраните текущую таблицу.

Щелкнув по нижней кнопке  в правом верхнем углу окна таблицы, закройте таблицу.

Повторите аналогичную операцию еще раз, в результате чего вы закроете текущую базу данных, получив пустое окно для новой работы. В этом положении можно создать новую базу данных, а можно открыть существующую для продолжения работы.


Откройте снова свою базу данных с помощью команд «**Файл**» > **Открыть** > **Студент.mdb** > **Открыть**»;

Откройте таблицу «**Список**».

**Задание 6****поиск записей по образцу**

Выполните следующие действия:

1. Установите текстовый курсор в поле «**Фамилия**»;

Щелкните по кнопке , которая позволяет найти запись по введенному значению. В результате появится диалоговое окно (рис. 5.21).

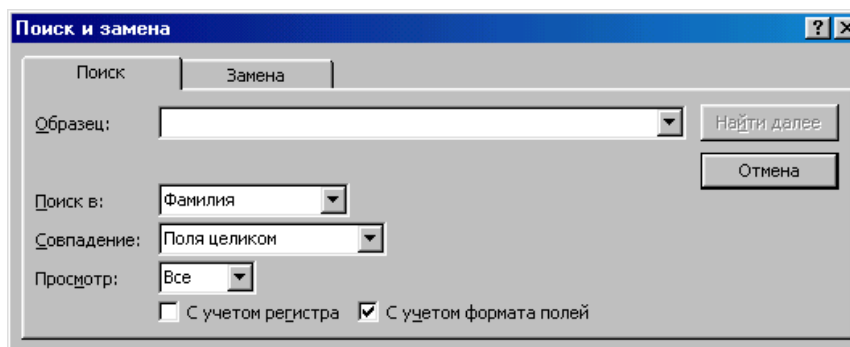


Рис.5.21. Поиск записей

Наберите в поле «**Образец**» фамилию «**Баранова**» и щелкните по кнопке «**Найти**». Если требуется найти следующую подобную запись, то щелкните мышкой по кнопке «**Найти далее**». Поиск осуществляется в текущем поле, если установлен соответствующий флажок. В противном случае поиск идет по всем полям. При этом можно учитывать регистр. По окончании работы щелкните по кнопке «**Закреть**».

Найдите студентов, родившихся в 1982 году.

**Задание 7****Печать таблицы**

(при отсутствии принтера задание не выполняется).

Выполните следующие действия:

1. Подготовьте принтер для печати.

Подготовьте таблицу для печати. Для этого щелкните по кнопке «**Предварительный просмотр**». Если таблица не уместилась на листе или видны не все данные, то закройте окно предварительного просмотра и поменяйте размер столбцов. Можно поменять поля листа или развернуть лист поперек (это можно сделать по команде «**Параметры страницы**» > **Страница** > **Альбомная** > **ОК**»).

Выполните команду «**Файл**» > **Печать...**». Появится окно (рис. 5.10). В этом окне возможен выбор варианта печати. Вы можете выбрать печать: всей таблицы, только выде-

ленных записей, только определенных страниц. Можно задать печать нескольких копий данного документа одной командой. Если к вашему компьютеру подключены несколько разных принтеров, то в поле «*имя*» можно выбрать нужный принтер.

Оставьте включенным переключатель «*Все*».

Щелкните по кнопке «*ОК*».

### Задание 8

#### Завершение работы с Access

Выполните следующие действия:

1. Выберите пункт меню «*Файл > Выход*».

Если вы производили какие-либо действия в базе данных, появится вопрос о сохранении изменений. Ответьте утвердительно на этот вопрос.

### 5.3. СОЗДАНИЕ БАЗЫ ДАННЫХ, СОСТОЯЩЕЙ ИЗ ДВУХ ТАБЛИЦ

#### Цели работы:

- 1) научиться удалять таблицы;
- 2) научиться создавать таблицы базы данных в режиме таблицы;
- 3) научиться создавать самостоятельно ключевое поле;
- 4) научиться создавать формы для ввода данных;
- 5) закрепить навыки по заполнению и редактированию таблиц базы данных.

#### Этапы работы

Создание базы данных осуществляется поэтапно:

1) **проектирование структуры базы данных.** Создадим более сложный вариант базы данных. В ней будут две таблицы: «*Список*» и «*Группы*»;

2) **конструирование пустых таблиц базы данных.** Воспользуемся новым способом изготовления таблиц. Таблицы будем создавать **в режиме таблицы**. В таблице «*Список*» будет 7 полей (код, фамилия, имя, отчество, год рождения, факультет, группа). Группы и фамилии кураторов групп будут храниться в отдельной таблице «*Группы*» в виде двух столбцов;

3) **создание схемы базы данных.** В данном случае таблицы связаны связью «*один-ко-многим*». Это значит, что в таблице «*Группы*» каждое значение может встречаться только один раз, а в таблице «*Список*» – сколько угодно (несколько человек могут быть из одной группы). Связи следует устанавливать при пустых таблицах. Если таблицы заполнены, могут возникнуть проблемы при создании связей и свойств связей. Для связи в обеих таблицах должны быть ключевые поля. В таблице «*Список*» – поле «*Код*», в таблице «*Группы*» – поле «*Группа*»;

4) **ввод данных в таблицы.** Создадим форму для ввода данных и воспользуемся ею. При наличии связанных таблиц имеет значение порядок заполнения их значениями.

### Задание 1

#### Открытие базы данных

Откройте учебную базу данных, спроектированную на прошлом занятии.

1. Запустите программу *Access*.

Выберите из списка баз данных имя «*Студент.mdb*». На экране появится главное окно базы данных (рис. 5.23) с основными элементами базы данных.

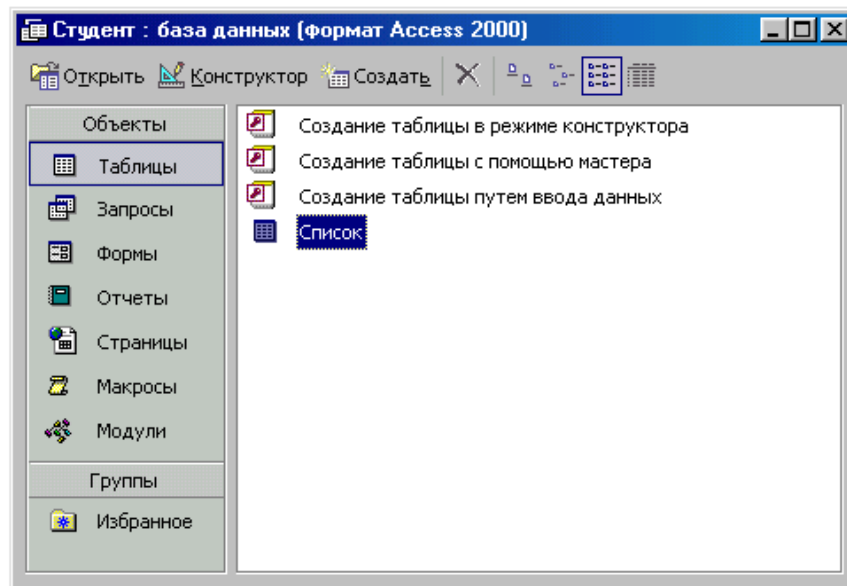


Рис. 5.23. Открытие базы данных

**Задание 2****Удаление таблицы**

Удалите таблицу «Список», созданную ранее.

1. Выделите таблицу «Список».
2. Нажмите клавишу «Delete».
3. На вопрос о подтверждении удаления таблицы ответьте «Да».

**Задание 3****Создание таблицы «группы»**

Для создание таблицы «Группы» выполните следующие действия:

1. Выберите закладку «Таблица». Щелкните мышкой по кнопке «Создать».
- Появится окно (рис. 5.24).

Оставьте «Режим таблицы» и щелкните по кнопке «ОК». Появится пустая таблица, поля которой не определены и не имеют названия. Тип поля будет выбран автоматически в зависимости от введенной информации.

Переименуйте «Поле 1». Для этого поставьте курсор в любую ячейку столбца «Поля 1». Выполните команду «Формат > Переименовать столбец». Ячейка имени столбца окажется выделенной. Введите название поля «Группа» и нажмите клавишу «Enter».

Переименуйте «Поле 2». Для этого поставьте курсор в любую ячейку столбца «Поля 2». Выполните команду «Формат > Переименовать столбец». Ячейка имени столбца окажется выделенной. Введите название поля «Куратор» и нажмите клавишу «Enter».

6. Сохраните таблицу с именем «Группы». На вопрос о создании ключевого поля ответьте отрицательно.

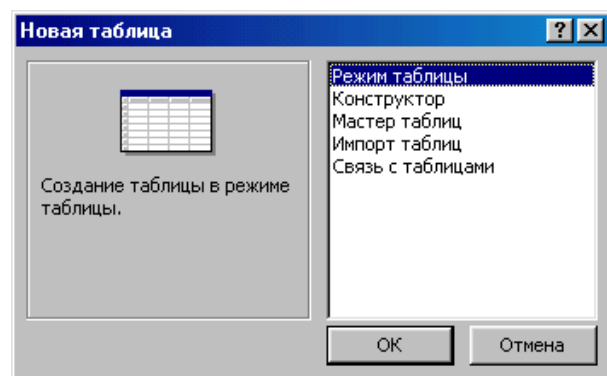




Рис. 5.24. Создание таблицы в режиме таблицы

7. Перейдите в режим **«Конструктор»**, щелкнув по кнопке , и посмотрите, как заданы поля. Сделайте поле **«Группа»** ключевым, поместив курсор на имя этого поля и щелкнув по кнопке  (**Ключевое поле**).

8. Сохраните таблицу. Закройте таблицу (при сохранении таблицы вопросов не появится, так как имя таблицы уже задано).

#### Задание 4

##### создание таблицы «Список»

Для создание таблицы **«Список»** выполните следующие действия:

1. Выберите закладку **«Таблица»**.
2. Щелкните мышью по кнопке **«Создать»**.
3. Появится окно, в котором оставьте **«Режим таблицы»** и щелкните мышью по кнопке **«ОК»**. Появится пустая таблица, поля которой не определены и не имеют названия.
4. Переименуйте поля по аналогии с предыдущим заданием, выполнив команду **«Формат > Переименовать поле»**. Данные возьмите из табл. 5.3.

Таблица 5.3

Данные для создания таблицы «Список»

Старое название	Новое название
Поле1	Код
Поле2	Фамилия
Поле3	Имя
Поле4	Отчество
Поле5	Год рождения
Поле6	Группа

5. Сохраните таблицу с именем **«Список»**. На вопрос о создании ключевого поля ответьте отрицательно.

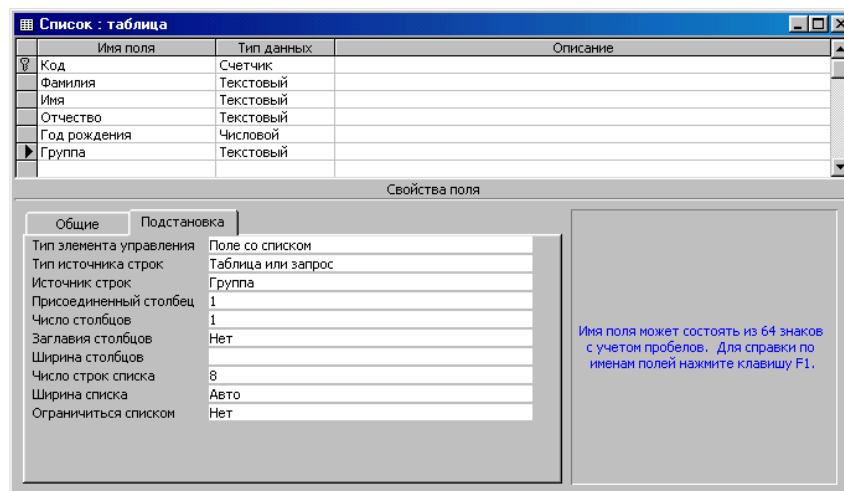



Рис. 5.25. Структура таблицы

6. Перейдите в режим **«Конструктор»** и посмотрите, как заданы поля. Сделайте поле **«Код»** ключевым, поместив курсор на имя этого поля и нажав кнопку  (**Ключевое поле**). Тип поля **«Код»** – счетчик, полей **«Фамилия»**, **«Имя»**, **«Отчество»** - текстовые, поля **«Год рождения»** – числовой. Поле **«Группа»** имеет особенности. Его тип – текстовый. Общие свойства поля не меняем. Кроме того, значения этого поля надо не набивать вручную, а выбирать из списка, содержащегося в таблице **«Группы»**. Получите значения элементов, как показано на рис. 5.25.


7. Сохраните изменения.
8. Закройте таблицу.

В результате этой работы вы получите две несвязанные таблицы. Чтобы установить связи между таблицами, необходимо создать схему данных со связями.

### Задание 5

#### Создание схемы данных

Для создания схемы данных выполните следующее:

1. Щелкните по кнопке  (*Схема данных*).
2. В появившемся окне «*Добавление таблицы*» выделите таблицу «*Группы*» и щелкните по кнопке «*Добавить*». Выделите таблицу «*Список*» и щелкните по кнопке «*Добавить*». В окне «*Схема данных*» появится условный вид этих таблиц. Щелкните по кнопке «*Закреть*» окна «*Добавление таблицы*».

3. Увеличьте окно таблицы «*Список*» так, чтобы были видны все поля.

4. Поставьте мышку на имя поля «*Группа*» в таблице «*Группы*» и, не отпуская кнопку мышки, перетащите ее на поле «*Группа*» в таблице «*Список*». Отпустите мышку. Появится диалоговое окно «*Изменение связей*» (рис. 5.26.)

5. Включите значок «*Обеспечение целостности данных*». Это невозможно будет сделать, если типы обоих полей заданы не одинаково.

6. Включите значок «*Каскадное обновление связанных полей*». Это приведет к тому, что при изменении группы в таблице «*Группы*» автоматически изменится соответствующая группа в таблице «*Список*».

7. Включите значок «*Каскадное удаление связанных полей*». Это приведет к тому, что при удалении записи с группой в таблице «*Группы*» будут удалены все записи из таблицы «*Список*», в которой стояли соответствующие группы.

8. Щелкните по кнопке «*Создать*». Появится связь «*один-ко-многим*». Схема данных представлена на рис. 5.27.

9. Закройте схему данных, утвердив ответив на вопрос о сохранении схемы данных.

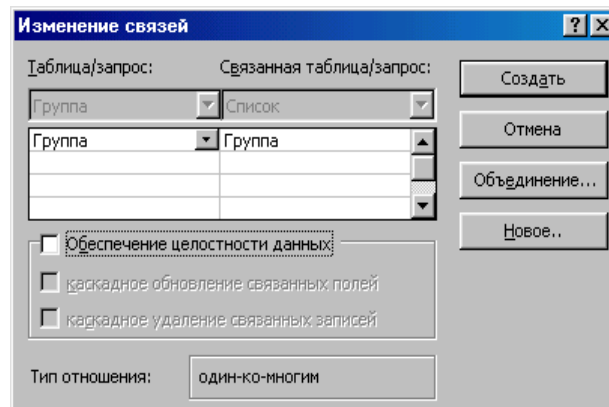


Рис. 5.26. Изменение связей

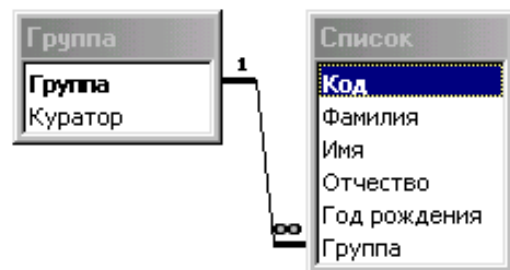


Рис. 5.27. Создание связи «*Один-ко-многим*»

### Задание 6

#### Заполнение таблицы «Группы»

Заполните таблицу «*Группы*» значениями.

1. Откройте таблицу «*Группы*» в режиме таблицы.
2. Заполните ее данными из табл. 5.4.
3. Сохраните таблицу и закройте ее.

Таблица 5.4

Данные для заполнения таблицы «Группы»

Группа	Преподаватель
0ТС	Верзаков С.А.
1МС	Белоусов А.И.
1ТМ	Масалова В.А.
2ИС	Новикова Е.В.

**Задание 7****Создание формы для ввода данных**

Для создания формы для ввода данных выполните следующее:

1. Выберите закладку «**Формы**».
2. Щелкните по кнопке «**Создать**».
3. Появится диалоговое окно (рис. 5.28), в котором следует выбрать «**Автоформа в столбец**», а в качестве источника данных – «**Список**».

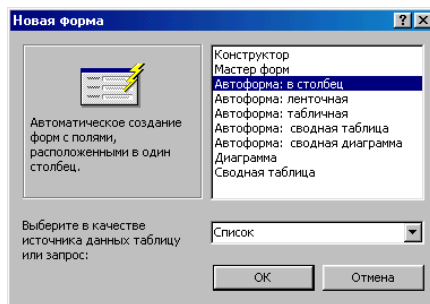


Рис. 5.28. Создание формы

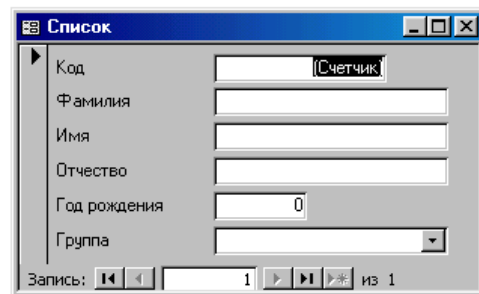


Рис. 5.29. Форма для ввода данных

4. Щелкните по кнопке «**ОК**». Появится пустая форма ввода (рис. 5.29).

Значения кнопок работы с записями:

- – переход к первой записи;
- – переход к предыдущей записи;
- – переход к следующей записи;
- – переход к последней записи;
- – добавление новой записи.

**Задание 8****Добавление записей**

Добавьте в базу данных записи, используя форму (см. рис. 5.29).

1. Заполните базу данными, представленными в табл. 5.5. Переход от поля ввода к следующему полю производится клавишей «**Tab**» или «**Enter**», или мышкой. Для перехода к новой записи следует щелкнуть по кнопке . Значения поля «**Код**» будут заполняться автоматически. Значения поля «**Группа**» выбирайте из ниспадающего списка.

Таблица 5.5

Заполненная база данных

Код	Фамилия	Имя	Отчество	Год рождения	Группа
1	Чернова	Кристина	Ивановна	1984	1МС
2	Терещенко	Инна	Алексеевна	1983	0ТС
3	Истратов	Максим	Владимирович	1984	1МС
4	Бондарь	Ольга	Петровна	1983	0ТС
5	Новоселов	Алексей	Иванович	1984	1ТМ

2. Сохраните введенные данные. Имя формы – «*Список*». Закройте форму.
3. Перейдите в окно «*Таблицы*». Откройте таблицу «*Список*». Убедитесь, что в таблице появились новые записи.
4. Сохраните текущую таблицу.
5. Закройте таблицу.

### Задание 9

#### Каскадное обновление связанных полей

Проверьте каскадное обновление связанных полей.

1. Откройте таблицу «*Группы*».
2. Исправьте учебные группы на 0ТС1, 1МС1, 1ТМ1, 2ИС1.
3. Сохраните таблицу.
4. Закройте таблицу «*Группы*».
5. Откройте таблицу «*Список*».
6. Убедитесь, что значения групп изменились.
7. Закройте таблицу «*Список*».

### Задание 10

#### Каскадное удаление связанных полей

Проверьте каскадное удаление связанных полей.

1. Откройте таблицу «*Группы*».
2. Удалите первую запись (выделите всю строку и нажмите на клавишу «*Delete*»).
3. Согласитесь с проверочным вопросом.
4. Закройте таблицу «*Группы*».
5. Откройте таблицу «*Список*».
6. Убедитесь, что исчезли записи с номером группы *0ТС1*.
7. Закройте таблицу «*Список*».

### Задание 11

#### Завершение работы с программой Access

Для завершения работы с программой Access выполните команду «*Файл > Выход*».

Если вы производили редактирование в базе данных, появится вопрос о сохранении изменений. Ответьте на него утвердительно.

## 5.4. СОЗДАНИЕ БАЗЫ ДАННЫХ, СОСТОЯЩЕЙ ИЗ ТРЕХ ТАБЛИЦ

### Цели работы:

- 1) научиться создавать таблицу базы данных с помощью *Мастера таблиц*;
- 2) закрепить навыки по добавлению и удалению записей;
- 3) закрепить навыки по заполнению и редактированию таблиц базы данных;
- 4) научиться использовать фильтр в таблице.

### Этапы работы:

- 1) *проектирование структуры базы данных*. Создадим более сложный вариант базы данных. В ней будет три таблицы: «*Список*», «*Группы*» и «*Личные данные*»;
- 2) *конструирование пустых таблиц базы данных*. Воспользуемся еще одним способом изготовления таблиц. Новую таблицу «*Личные данные*» создадим с помощью *Мастера таблиц*. Структуру таблиц «*Список*» и «*Группы*» скопируем из ранее составленной базы данных;

- 3) **создание схемы базы данных.** В данном случае таблицы «*Группы*» и «*Список*» объединены связью «один-ко-многим», таблицы «*Список*» и «*Личные данные*» – связью «один-к-одному». Таблицы «*Группы*» и «*Личные данные*» прямо не связаны;
- 4) **ввод данных в таблицы.** Создадим форму для ввода данных и воспользуемся ею.
- 5) **использование базы данных для практических задач.**

### Задание 1

#### Открытие базы данных

Откройте учебную базу данных, созданную ранее.

1. Запустите программу **Access**.
2. Откройте базу данных «*Студент.mdb*». Используем готовые таблицы этой базы для конструирования новой. Таблицу «*Группа*» оставим в прежнем виде, исправив испорченные значения, а из таблицы «*Список*» возьмем только структуру.

### Задание 2

#### Корректировка данных в таблице

Откорректируйте данные в таблице «*Группы*».

1. Ранее вы удаляли записи из таблицы «*Группы*». Их следует восстановить.
2. Откройте таблицу «*Группы*», выделив ее и щелкнув по кнопке «*Открыть*».
3. Добавьте недостающие записи, исправьте существующие. Таблица должна выглядеть следующим образом (табл. 5.7).

Таблица 5.7

Группа

Группа	Преподаватель	Группа	Преподаватель
0ТС1	Верзаков С.А.	2ИС1	Новикова Е.В.
1МС1	Белоусов А.И.	0МС1	Белоусов А.И.
1ТМ1	Масалова В.А.		

4. Закройте таблицу, предварительно сохранив ее.

### Задание 3

#### Удаление записей

Удалите все записи таблицы «*Список*», оставив ее структуру.


1. Откройте таблицу «*Список*».
2. Выделите все записи.
3. Нажмите клавишу «*Delete*». Щелкните по кнопке «*Да*» в вопросе о подтверждении удаления.
4. Закройте таблицу, сохранив ее. Если теперь вводить данные в эту таблицу снова, то счетчик будет меняться с того номера, который был присвоен последней записи.
5. Чтобы нумерация снова начиналась с единицы, выполните команду «*Сервис > Службные программы > Сжать*» и восстановить базу данных ....

### Задание 4

#### Создание таблицы «личные данные»

Используя *Мастер таблиц*, создайте таблицу «*Личные данные*» с ключевым полем.

1. Выберите закладку «*Таблица*».
2. Щелкните по кнопке «*Создать*». В результате перейдем к работе со следующим диалоговым окном: «*Новая таблица*». Здесь несколько вариантов, выберите «*Мастер таблиц*» и щелкните по кнопке «*ОК*». Появится диалоговое окно (рис. 5.30).
3. В этом окне следует выбрать: в поле «*Образцы таблиц*» – поле «*Студенты*»; в поле «*Образцы полей*» – поля «*КодСтудента*», «*Адрес*», «*НомерТелефона*»,

щелкая после каждого выбора по кнопке . Эти поля попадут в «Поля новой таблицы». Щелкните по кнопке «Далее».

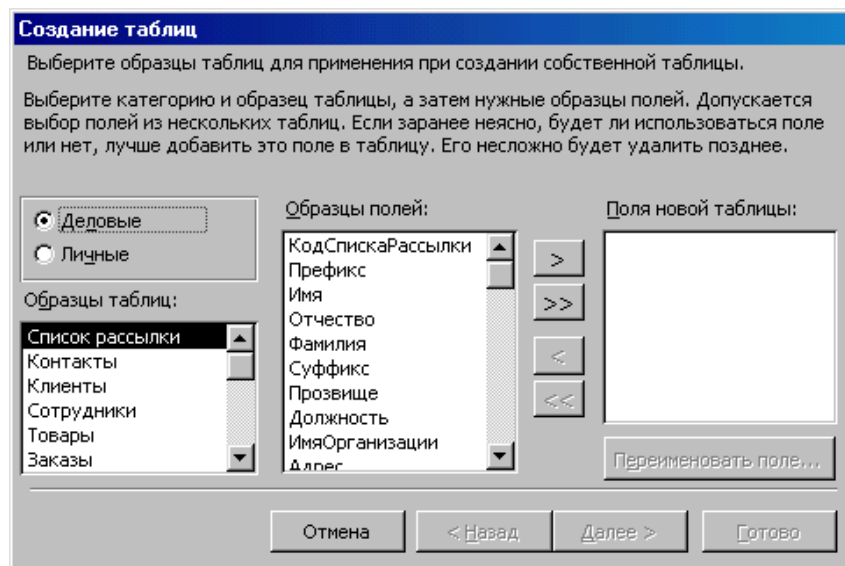


Рис. 5.30. Создание таблицы

4. В диалоговом окне задайте имя новой таблицы «**Личные данные**». Оставьте автоматический выбор ключа. Щелкните по кнопке «Далее». Access проверит связи данной таблицы с другими таблицами. Так как вы еще не устанавливали связи, то они не будут найдены автоматически. В этот момент можно установить новые связи, но мы пока этого делать не будем.

5. Щелкните по кнопке «Далее».

6. После появления вопроса о действиях после создания таблицы автоматически выбирайте ввод в режиме таблицы, но можно изготовить и форму. Щелкните по кнопке «Готово». Вы попадете в пустую таблицу, у которой есть поля, но отсутствуют записи.

7. Добавьте в таблицу «**Личные данные**» еще три поля «**Математика**», «**Физика**» и «**Химия**», в которых будут находиться семестровые оценки по этим предметам. Выберите «**Конструктор**». Добавьте в конец списка полей три поля с именами «**Математика**», «**Физика**», «**Химия**» и типом данных – «**числовой**». Измените тип поля «**КодСтудента**» на числовой.

8. Щелкните по кнопке «**Сохранить**».


9. Перейдите в режим таблицы.


10. Закройте таблицу, предварительно сохранив ее. В результате вы получите три таблицы, две из которых связаны, а третья нет.

## Задание 5

### Исправление схемы данных

Исправьте схему данных.

1. Щелкните по кнопке  («**Схема данных**»). Появится диалоговое окно «**Схема данных**».

2. Щелкните по кнопке  («**Добавить таблицу**»). В появившемся окне «**Добавление таблицы**» выделите таблицу «**Личные данные**» и щелкните по кнопке «**Добавить**», а затем – по кнопке «**Закреть**» окна «**Добавление таблицы**».

3. Поставьте мышку на имя поля «**КодСтудента**» в таблице «**Личные данные**» и, не отпуская кнопку мышки, перетащите ее на поле «**Код**» в таблице «**Список**». Отпустите мышку. Появится диалоговое окно «**Изменение связей**».

4. Включите флажок «**Обеспечение целостности данных**». Это невозможно сделать, если типы обоих полей заданы неодинаково.

5. Щелкните по кнопке «**Создать**». Появится связь «один-к-одному». Это значит, что одной записи в таблице «**Список**» соответствует одна запись в таблице «**Личные данные**». Схема данных представлена на рис. 5.31.



Рис. 5.31. Создание связи «Один-к-одному»

6. Закройте схему данных, сохранив ее. Теперь встает задача ввода записей одновременно в разные таблицы: «**Список**» и «**Личные данные**». **Фамилии, имена и отчества** (ФИО) мы храним в одной таблице, а **адрес** и **номер телефона** – в другой. Можно, конечно, попробовать ввести значения в каждую таблицу отдельно, но удобнее видеть клетки обеих таблиц для ввода данных одновременно. Эту задачу легко решить вводом значений через специально созданную форму, в которой присутствуют поля всех необходимых таблиц. Данные вводим в форму, а в результате заполняются таблицы.

## Задание 6

### Создание формы для ввода данных

Создайте форму для ввода данных.

Форма представляет собой специальный формат экрана, используемый для разных целей, чаще всего для ввода данных в таблицу и просмотра одной записи. Формы позволяют вводить данные, корректировать их, добавлять и удалять записи. Можно создавать формы для работы одновременно с несколькими взаимосвязанными таблицами. Форма, использующая данные из нескольких таблиц, должна быть основана на запросе, включающем данные из этих таблиц.

С применением форм можно представлять записи в удобном для пользователя виде – в виде привычных документов: бланков, экзаменационных ведомостей и т.д. Формы ввода-вывода позволяют вводить данные в базу, просматривать их, изменять значения полей, добавлять и удалять записи.



Все элементы, добавляемые в форму (поля, надписи, списки, переключатели, кнопки, линии), являются элементами управления. Способ создания элемента управления зависит от того, какой элемент создается: присоединенный, свободный или вычисляемый.

Для создания формы для ввода данных выполните следующее:

1. В окне базы данных выберите вкладку «**Формы**».

2. Щелкните по кнопке «**Создать**».

3. В диалоговом окне «**Новая форма**» выберите «**Мастер форм**». Описание **Мастера** появляется в левой части диалогового окна. В нижнем поле имя таблицы или запроса в качестве источника данных можно не указывать. При использовании **Мастера форм** источник данных для формы следует указывать в диалоговом окне **Мастера**.

4. Щелкните по кнопке «**ОК**». Появится окно создания форм (рис. 5.32).
5. Выберите все поля из таблицы «**Список**» и все поля – из таблицы «**Личные данные**». Для этого выберите имя таблицы «**Список**» в поле «**Таблицы и запросы**». В результате появляется список полей в окне «**Доступные поля**». Щелкните по кнопке , которая переносит все поля из списка. Затем выберите имя таблицы «**Личные данные**» в поле «**Таблицы и запросы**» и вновь щелкните по кнопке .

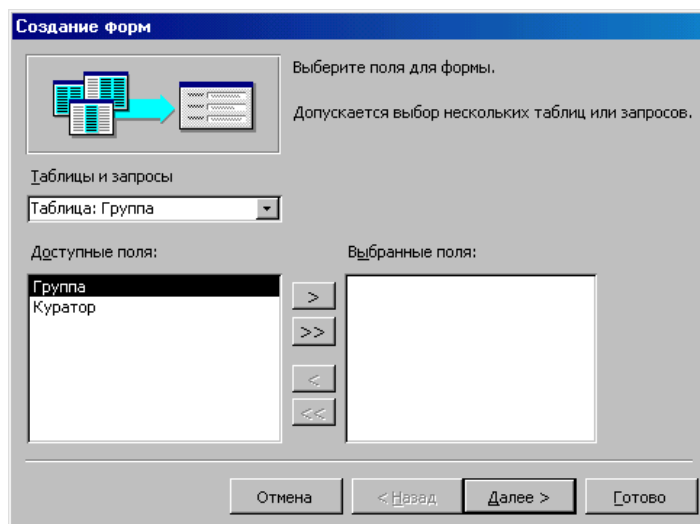


Рис. 5.32. Создание формы с помощью Мастера

6. Щелкните по кнопке «**Далее**».
7. Оставьте внешний вид формы в один столбец (выбран по умолчанию).
8. Щелкните по кнопке «**Далее**».
9. Выберите требуемый стиль (например, «**Обычный**»). Щелкните по кнопке «**Далее**».
10. Задайте имя формы: «**Общая форма**». Щелкните по кнопке «**Готово**». В результате вы получите форму (рис. 5.33), в которой можно менять существующие данные и вводить новые значения. Эти значения будут попадать в ту таблицу, в которую нужно (часть значений – в одну таблицу, часть – в другую).

Рис. 5.33. Готовая форма

**Задание 7****Заполнение таблиц данными**

Поля «Код» и «КодСтудента» заполняются автоматически. Далее выполните следующее:

1. Заполните таблицы данными: данные таблицы «Список» приведены в табл. 5.8, а таблицы «Личные данные» – в табл. 5.9.

Таблица 5.8

Данные для заполнения таблицы «Список»

Код	Фамилия	Имя	Отчество	Год рождения	Группа
1	Иванова	Анна	Ивановна	1984	1МС
2	Баранова	Ирина	Алексеевна	1983	0ТС
3	Корнилова	Ольга	Владимировна	1984	1МС
4	Воробьев	Алексей	Петрович	1983	0ТС
5	Воробьев	Алексей	Иванович	1984	1МС
6	Воробьев	Олег	Григорьевич	1985	2ТМ
7	Скоркин	Александр	Евгеньевич	1982	0ТС
8	Володина	Анна	Алексеевна	1984	1МС
9	Новоселов	Алексей	Антонович	1983	0ТС
10	Александрова	Елена	Алексеевна	1984	1МС

Таблица 5.9

Данные для заполнения таблицы «Личные данные»

Код студента	Адрес	Номер телефона	Математика	Физика	Химия
1	Центральная 11-5	51-17-22	5	5	5
2	Солнечная 8-117	51-18-22	4	4	4
3	Сиреневый 7-16	51-19-22	3	4	5
4	Центральная 14-81	51-20-22	5	5	4
5	Сиреневый 7-16	51-21-22	5	4	4
6	Солнечная 2-121	51-22-22	4	4	4
7	Школьная 5-34	51-23-22	5	5	5
8	Центральная 30-7	51-24-22	3	3	3
9	Сиреневый 7-16	51-25-22	3	4	4
10	Солнечная 6-34	51-26-22	5	5	5

2. Закройте форму, предварительно сохранив ее.
3. Перейдите на закладку «Таблицы».
4. Откройте таблицу «Список» и убедитесь, что в них появились данные. Закройте таблицу.
5. Откройте таблицу «Личные данные» и убедитесь, что в них появились данные. Закройте таблицу.

**Задание 8****Добавление поля «портрет»**

Добавьте новое поле «Портрет» (рисунки или фотографии) в таблицу «Список». Возможен вариант, когда появляется необходимость добавить новые поля в существующую таблицу. Это можно сделать, например, таким способом:

1. Откройте таблицу «Список».
2. Перейдите в режим «Конструктора».

3. Добавьте еще одно поле (вводите имя поля ниже поля «*Группа*»), имя которого «*Портрет*», тип данных – «*Поле объекта OLE*», общие свойства поля оставить по умолчанию.
4. Щелкните по кнопке «*Сохранить*».
5. Перейдите в режим таблицы.
6. Щелкните мышкой по клетке, где должно быть значение поля «*Портрет*».
7. Выполните команду «*Вставка > Объект > Точечный рисунок > ОК*».
8. Нарисуйте портрет.
9. Закройте окно рисунка, в результате чего вы вернетесь в таблицу. Рисунок будет обозначен словами. Чтобы увидеть портрет, дважды щелкните мышкой по названию рисунка, возвращаясь в программу, где изготовлен портрет.
10. Добавьте фото для каждого студента.

### Задание 9

#### Использование фильтра


Выражения в фильтре могут состоять из точных значений, которые Access использует для сравнения в том виде, в котором они вводятся. Числа вводятся без ограничителей, например, **22**. Текст должен быть заключен в кавычки, например «*Александров*». Даты ограничиваются символами #, например:

#10/01/99#.

Элементы выражения могут быть связаны операторами:

- арифметическими (\*, +, -, /, ^);
- сравнения(<, <=, >, >=, =, <>);
- логическими – *And* (И), *Not* (Нет), *Or* (Или);
- *Like* – для использования логики замены в выражениях;
- *In* – для определения, содержится ли элемент данных в списке значений;
- *Between...And* – для выбора значений из определенного интервала.

Научитесь использовать фильтр.

1. Щелкните по кнопке  («*Изменить фильтр*»). Появится окно выбора (рис. 5.34).
2. Щелкните мышкой по полю «*Год рождения*». У активного поля появится стрелка выбора.

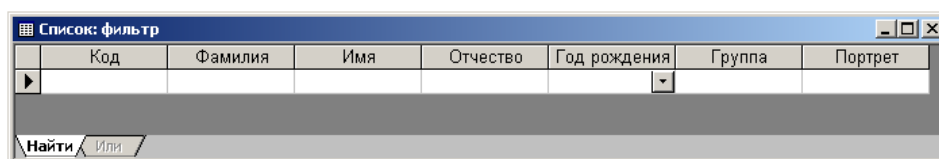





Рис. 5.34. Использование фильтра

3. Выберите «*Год рождения* 1984 и щелкните по кнопке  («*Применить фильтр*»). Вы автоматически попадете в таблицу, в которой будут только выбранные записи.
4. Отмените выбор. Для этого необходимо отжать эту же кнопку, которая теперь называется «*Удалить фильтр*».
5. Щелкните по кнопке  («*Изменить фильтр*»).
6. Удалите все в поле «*Год рождения*», выделив значение и нажав клавишу «*Delete*».

7. Измените фильтр так, чтобы в таблице были видны только атрибуты студентов группы ОТС, 1983 года рождения (одновременный запрос в двух полях – «*Группа*» и «*Год рождения*»).

8. Щелкните по кнопке  («*Применить фильтр*»).



9. Измените фильтр. Допустимо указывать границы изменения значений. В поле «*Год рождения*» наберите «>1983». Щелкнув по кнопке , вы получите таблицу, в которой присутствуют записи с годами рождения больше 1983.

10. Чтобы получить записи студентов, у которых фамилии начинаются на букву «*В*», в соответствующем поле наберите *Like «В\*»*. Составьте этот запрос.

11. Запрос *Not «В\*»* будет означать все записи, кроме указанных (в данном случае все записи, у которых фамилии не начинаются на букву «*В*»). Составьте этот запрос.

12. Самостоятельное задание:

- Выберите студентов всех групп, кроме группы *ИМС*.
- Выберите студентов группы *ИМС*, фамилии которых начинаются на букву «*В*».
- Выберите студентов, год рождения которых *1983* или *1984* (запрос: 1983 OR 1984).
- Отмените все запросы.

13. Кнопка  («*Фильтр по выделенному*») позволяет оставить видимыми в таблице только те записи, в которых есть предварительно выделенный элемент. В поле («*Фамилия*») выделите две первые буквы фамилии «*Володина*». Щелкнув по кнопке  («*Фильтр по выделенному*»), вы получите таблицу, в которой присутствуют записи с фамилиями, начинающимися на «*Во*».

### Задание 10

#### Завершение работы с программой Access

Завершите работу с программой Access.

1. Выберите пункт меню «*Файл > Выход*».
2. Если вы производили редактирование в базе данных, появится вопрос о сохранении изменений. Отвечайте на него утвердительно.

## 5.5. ПРИМЕНЕНИЕ ФОРМ

### Цели работы:

- 1) научиться создавать формы ввода-вывода;
- 2) научиться создавать кнопочные формы.

### Задание 1

#### Открытие базы

Откройте учебную базу данных, созданную ранее, и подготовьте ее к работе.

- 1) запустите программу Access;
- 2) выберите «*Открыть базу данных*», выделите из списка баз данных, расположенного ниже переключателя, имя вашей базы и щелкните по кнопке «*ОК*»;
- 3) перейдите на закладку «*Формы*», выделите форму «*Список*» и нажмите клавишу «*Delete*». Согласитесь с удалением. В результате у вас в базе данных должны остаться три исходные таблицы и «*Общая форма*», с помощью которой вы можете добавлять значения в вашу базу данных.

## Задание 2

### Создание простой формы

Создайте простую форму с помощью кнопки «*Новый объект*».

1. В диалоговом окне базы данных выберите вкладку «*Таблицы*».

2. Выделите таблицу «*Список*» (автоформу можно строить и при закрытой таблице, и при открытой). Автоформа создает форму, в которой отображаются все поля и записи выбранной таблицы или запроса. Каждое поле располагается на отдельной строке, с левой стороны от которой отображается надпись к данному полю. С помощью автоформы можно просматривать или вводить данные.

3. Щелкните по кнопке раскрытия списка, расположенной рядом с кнопкой «*Новый объект*» на панели инструментов и выберите элемент «*Автоформа*». Появится автоформа (рис. 5.35).

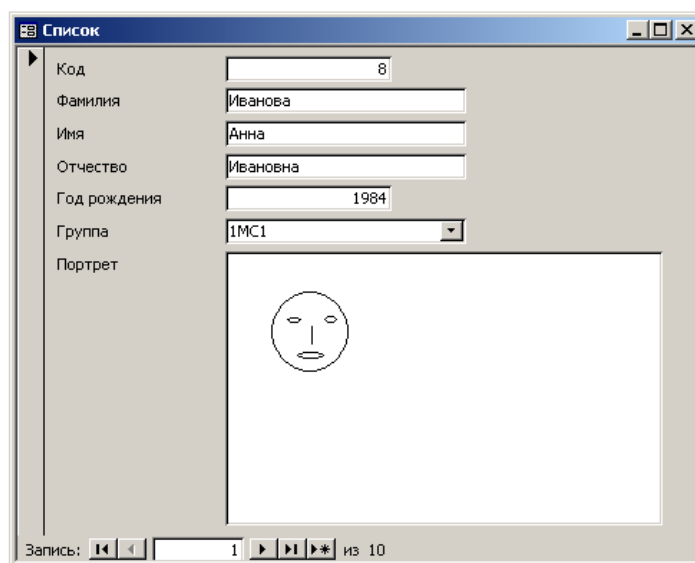


Рис. 5.35. Создание простой формы

4. Сохраните автоформу с именем «*Список*».

5. Закройте автоформу, ответив утвердительно на вопрос о сохранении, если таковой последует.

## Задание 3

### Создание формы с помощью мастера форм

Создайте форму с помощью *Мастера форм*.

1. В окне базы данных выберите вкладку «*Формы*».

2. Щелкните по кнопке «*Создать*».

3. В диалоговом окне «*Новая форма*» выберите «*Мастер форм*». Описание *Мастера* появляется в левой части диалогового окна. В нижнем поле имя таблицы или запроса в качестве источника данных можно не указывать. При использовании *Мастера форм* источник данных для формы следует указывать в диалоговом окне *Мастера*.

4. Щелкните по кнопке «*ОК*».

5. В поле «*Таблицы/запросы*» выберите таблицу «*Список*», в поле «*Доступные поля*» выберите поля «*Фамилия*», «*Имя*» и перенесите их стрелкой в поле «*Выбранные поля*».

6. В поле «**Таблицы/запросы**» выберите таблицу «**Личные данные**», в поле «**Доступные поля**» выберите поля «**Математика**», «**Физика**», «**Химия**» и перенесите их стрелкой в поле «**Выбранные поля**». Результат этих действий представлен на рис. 5.36.

7. Щелкните на кнопке «**Далее**».

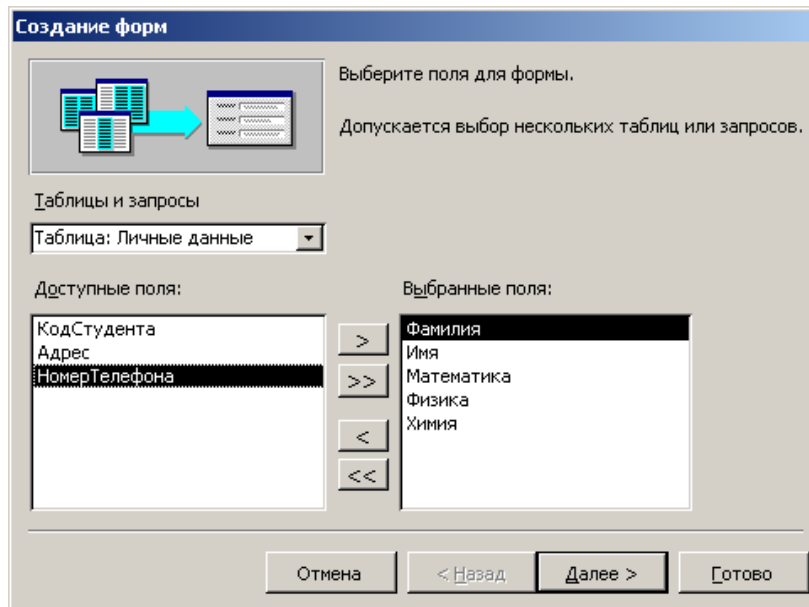


Рис. 5.36. Создание формы с помощью Мастера

8. Выберите внешний вид формы – «**Табличный**». Щелкните по кнопке «**Далее**».

9. Выберите требуемый стиль (например, «**Стандартный**»). Щелкните по кнопке «**Далее**».

10. Задайте имя формы: «**Успеваемость**». Щелкните по кнопке «**Готово**». В результате получите форму, в которой можно менять существующие данные и вводить новые значения сразу в две таблицы.

11. Закройте форму.

#### Задание 4

##### Создание формы с помощью автоформ

Создайте форму с помощью автоформ.

1. Щелкните по кнопке «**Создать**». При выборе элементов «**Автоформа: в столбец**», «**Автоформа: ленточная**» или «**Автоформа: табличная**» форма создается автоматически.

2. Выберите «**Автоформа: ленточная**».

3. В нижнем поле выберите таблицу «**Личные данные**», которые будут служить источником данных для формы.

4. Щелкните по кнопке «**ОК**». Получите готовую автоформу.

5. Сохраните ее с именем «**Личные данные**».

6. Закройте автоформу.


7. Самостоятельное задание:

- изготовьте автоформу в столбец для таблицы «**Список**» (Список учеников);
- изготовьте табличную автоформу для таблицы «**Группы**» (Номера групп).

8. Закройте формы, если они открыты.

**Задание 5****Создание формы без помощи «Мастера»**

Создайте формы без помощи «Мастера».

1. В диалоговом окне базы данных выберите вкладку «**Формы**», если находитесь в другом окне.
2. Щелкните по кнопке «**Создать**».
3. В диалоговом окне «**Новая форма**» выберите режим «**Конструктор**».
4. В открывшемся окне выберите таблицу «**Список**», содержащую данные, на основе которых будет создана форма.
5. Щелкните по кнопке «**ОК**».
6. Окно формы будет выведено на экран в режиме «**Конструктор**».
7. Щелкните по кнопке  («**Список полей**») на панели инструментов. Получите список полей, из которого можно добавлять присоединенные элементы управления в форму или в отчет. Выделите поля «**Фамилия**», «**Имя**», «**Отчество**», «**Группа**», «**Портрет**». Перетащите мышкой поля в область данных.
8. Закройте окно списка полей.

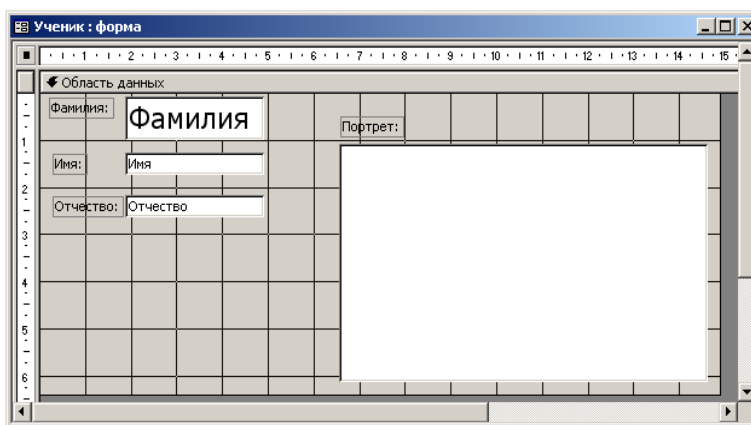


Рис. 5.37. Вывод формы в режиме Конструктора

9. Расположите элементы на форме как на рис. 5.37. Для независимого друг от друга перемещения элементов следует «захватить» их мышкой за левый верхний квадрат большего размера, иначе подпись и поле будут передвигаться вместе.

10. Задайте размер текста поля «**Фамилия**» равным 18. Чтобы увеличить размер элемента соответственно надписи, выполните команду «**Формат > Размер > по размеру данных**».

11. Сохраните форму с именем «**Ученик**».

12. Формы могут быть выведены на экран в трех видах: **режим конструктора** (рис. 5.37), **режим формы** (рис. 5.38) и **режим таблицы**. Переключение режимов производится кнопкой «**Вид**». Посмотрите все способы представления формы.

13. Закройте форму.

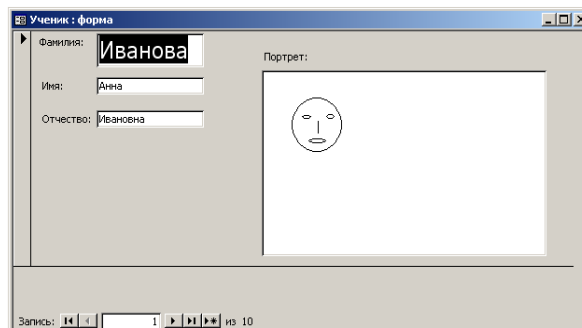


Рис. 5.38. Готовая форма

**Задание 6****Добавление логического поля**

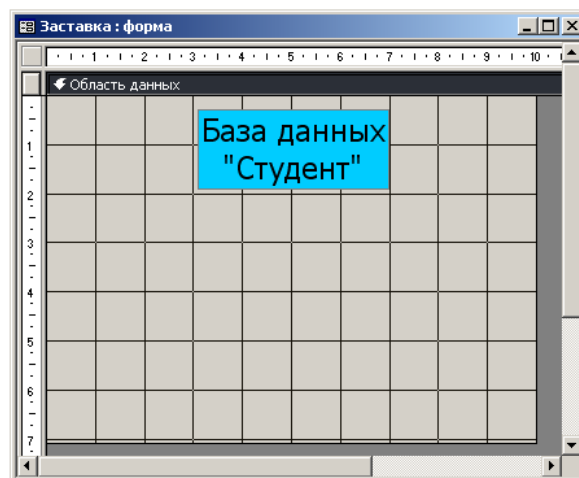
Добавьте в таблицу «*Список*» логическое поле «*Аспирантура*», значения которого «*Да*» или «*Нет*», которое будет содержать информацию о том, собирается ли студент поступать в аспирантуру. Добавьте это поле в форму.

1. Перейдите на закладку «*Таблицы*». Откройте таблицу «*Список*» в режиме «*Конструктор*». Добавьте поле с именем «*Аспирантура*» и типом «*Логический*». Сохраните таблицу и закройте ее.
2. Перейдите на закладку «*Формы*».
3. Откройте форму «*Ученик*» в режиме «*Конструктор*». В форме или в отчете флажок может быть использован как отдельный элемент управления, в котором отображаются значения логического поля из базовой таблицы. Если флажок установлен, поле в таблице имеет значение «*Да*», если флажок снят, поле имеет значение «*Нет*».
4. Щелкните по кнопке «*Список полей*». Выделите название «*Аспирантура*» и перетащите его мышкой в область данных. Появится значок флажка и надпись «*Аспирантура*».
5. Перейдите в режим формы и посмотрите разные записи. Флажок снимается и устанавливается щелчком мышки по соответствующему полю. Установите флажки у трех студентов.
6. Закройте форму, ответив утвердительно на вопрос о сохранении.


**Задание 7****Создание кнопочной формы (вариант 1)**

Создайте кнопочную форму «*Заставка*».

1. Щелкните по кнопке «*Создать*».
2. В появившемся диалоговом окне выберите «*Конструктор*». Появится пустая форма. Задайте мышкой ширину формы, равную 10 см, а высоту – 7 см.
3. Сохраните форму с именем «*Заставка*».
4. Выберите на панели инструментов кнопку («*Надпись*»). Курсор мышки примет вид крестика с «приклеенной» буквой «*А*». Щелкните мышкой по месту начала надписи и введите: «*База данных*» «*Студент*» (после слов «*База данных*» нажмите одновременно комбинацию клавиш «*Shift + Enter*»).
5. Нажмите клавишу «*Enter*».



6. Выберите размер букв **18**, а выравнивание – **по центру**. Выполните команду меню «*Формат > Размер > по размеру данных*». Цвет фона – **голубой**.
7. Растяните мышкой надпись на ширину окна, как представлено на рис. 5.39.

8. Выберите на панели элементов значок  («Кнопка»).
9. Щелкните мышкой по тому месту в области данных, где должна быть кнопка (это может быть любое место, которое вам нравится). Появится диалоговое окно «Создание кнопок».
10. Выберите категорию «Работа с формой», а действие – «Открыть форму» (рис. 5.40).

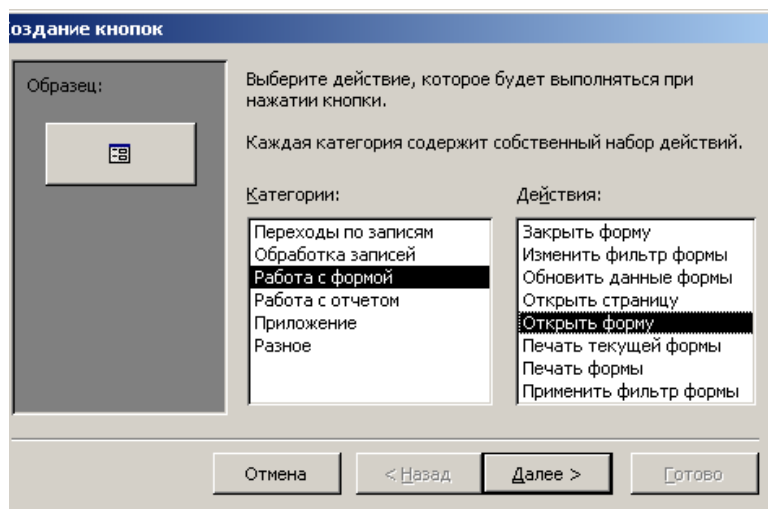


Рис. 5.40. Мастер создания кнопки

11. Щелкните по кнопке «Далее».
12. Выберите форму «Список», открываемую этой кнопкой, и щелкните по кнопке «Далее».
13. В следующем окне щелкните по кнопке «Далее».
14. В следующем окне поставьте переключатель в положение «Текст», наберите в поле «Текст» слово «Список», как показано на рис. 5.41.

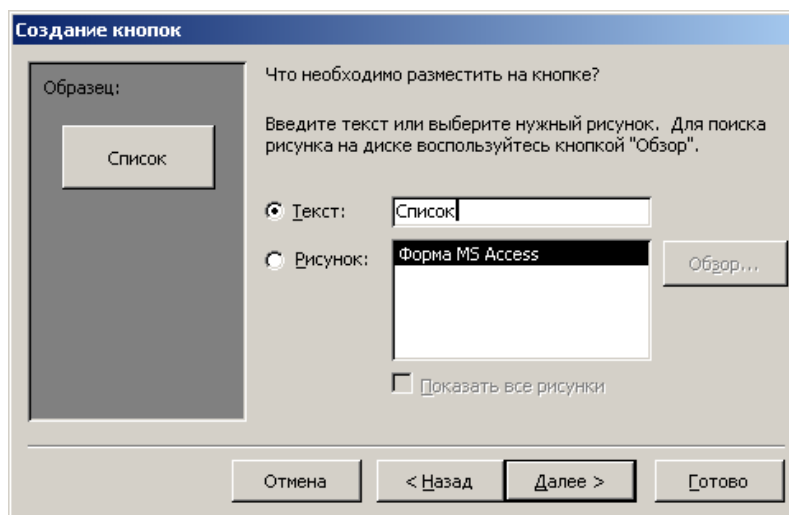


Рис. 5.41. Мастер создания кнопки

15. Щелкните по кнопке «Далее».
16. Задайте имя кнопки «Список» и щелкните по кнопке «Готово».
17. Самостоятельно создайте кнопки для форм «Успеваемость» и «Номера групп».

18. Перейдите в режим формы (рис. 5.42). Теперь при щелчке мышью по соответствующим кнопкам будут открываться соответствующие формы для работы.

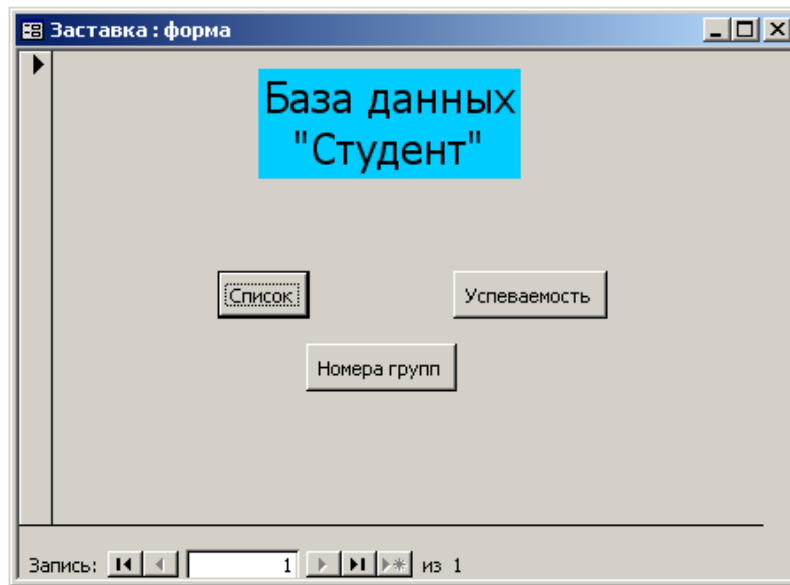


Рис. 5.42. Готовая кнопочная форма

19. Чтобы кнопочная форма автоматически появлялась при запуске базы данных, выполните команду «Сервис > Параметры запуска». В поле «Форма» выберите «Заставка» и щелкните по кнопке «ОК».

20. Закройте форму, ответив утвердительно на возможный вопрос о ее сохранении.

### Задание 8

#### Создание кнопочной формы (вариант 2)

Создайте кнопочную форму при помощи *Диспетчера кнопочных форм*.

1. Выполните команду «Сервис > Служебные программы > Диспетчер кнопочных форм», на заданный вопрос ответьте «Да». В результате вы получите диалоговое окно «Диспетчер кнопочных форм» (рис. 5.43).

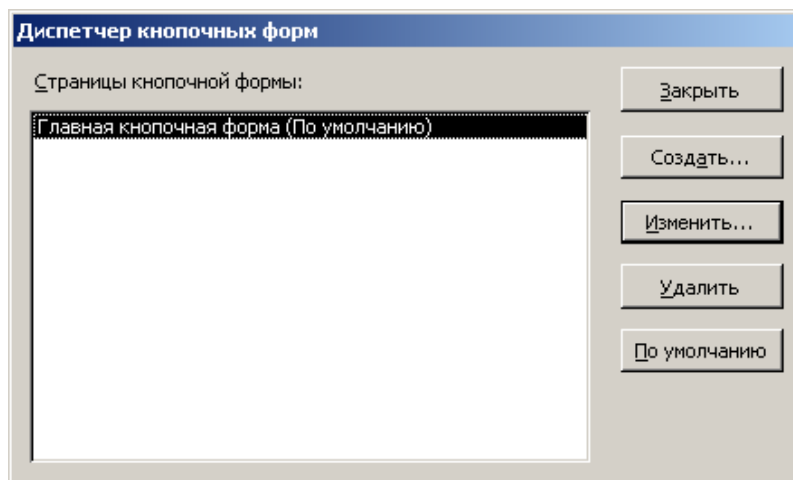


Рис. 5.43. Диспетчер кнопочных форм

2. Щелкните в окне «Диспетчер кнопочных форм» по кнопке «Изменить».

3. В открывшемся диалоговом окне **«Изменение страницы кнопочной формы»** щелкните по кнопке **«Создать»**. Появится диалоговая страница **«Изменение элемента кнопочной формы»**. Измените содержимое полей **«Текст»**, **«Команда»**, **«Форма»**, в соответствии с рис. 5.44.

Рис. 5.44. Изменение элемента кнопочной формы

4. Щелкните по кнопке **«ОК»**.
5. Аналогично в окне **«Изменение страницы кнопочной формы»** создайте еще два элемента кнопочной формы:
  - 1) **«Личные данные»**. Эта кнопка открывает форму **«Личные данные»** в режиме редактирования;
  - 2) **«Успеваемость»**. Эта кнопка открывает форму **«Успеваемость»** в режиме редактирования.
6. Добавьте кнопку закрытия базы данных. Для этого щелкните по кнопке **«Создать»**, наберите в поле **«Текст»** слово **«Выход»**, а в поле **«Команда»** выберите **«Выход из приложения»**. Закройте диалоговое окно **«Изменение страницы кнопочной формы»**, а затем – **«Диспетчер кнопочных форм»**.
7. Перейдите на закладку **«Формы»** и откройте окно **«Кнопочная форма»** в режиме **Конструктора**, измените цвет надписи, сохраните форму.
8. Перейдите в режим формы. Результат представлен на рис. 5.45.

Рис. 5.45. Готовая кнопочная форма

9. Проверьте работу всех кнопок кнопочной формы (после нажатия кнопки **«Выход»** снова откройте базу данных).

### Задание 9

#### Завершение работы с программой

Завершите работу с программой *Access*.

1. Выберите пункт меню **«Файл > Выход»**.
2. Если вы производили редактирование в базе данных, появится вопрос о сохранении изменений. Ответьте на него положительно.

## 5.6 СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ЗАПРОСОВ

### Цели работы

- 1) закрепить навыки по редактированию таблиц;
- 2) познакомиться с основными видами запросов;
- 3) научиться создавать запросы на выборку различными способами;
- 4) научиться создавать запросы: на обновление, на добавление, на удаление, на создание таблицы;
- 5) научиться создавать перекрестные запросы.

Предположим, что вы хотите создать телефонную книгу или получить список всех студентов, занимающихся у определенного преподавателя, используя конкретную базу данных. Подобную работу легко сделать с использованием запросов.

Предположим, что группу 1МС1 переименовали в 1СС1, а куратор группы Беловусов А.И. уволился и его сменил Иванов А.С., прошел год, некоторые студенты закончили университет, пришли новые. Нужно обновить базу данных.

### Задание 1

#### Открытие базы

Откройте учебную базу данных, изготовленную ранее.

1. Запустите *Access*.
2. Выберите пункт **«Открыть базу данных»**, выделите из списка баз данных, расположенного ниже переключателя, имя вашей базы и щелкните по кнопке **«ОК»**.
3. Если вы включили в параметры запуска **«Заставку»**, то она появится перед вами. Закройте ее. На экране возникнет диалоговое окно с основными элементами базы данных. В базе данных должны быть три таблицы: **«Список»**, **«Личные данные»**, **«Группы»**.

### Задание 2

#### Создание запроса

Запрос – это инструмент для анализа, выбора и изменения данных. С помощью запросов можно просматривать, анализировать и изменять данные из нескольких таблиц. Запросы используются также в качестве источника данных для форм и отчетов.

С помощью *Access* могут быть созданы несколько видов запросов:

- **«Запрос на выборку»** выбирает данные из разных таблиц и других готовых запросов;
- **«Запрос-изменение»** изменяет или перемещает данные; к этому типу относятся **«Запрос на добавление»**, **«Запрос на удаление»** и **«Запрос на обновление»**;
- **«Запрос на создание таблицы»** сохраняет результаты выборки в отдельной таблице;
- **«Перекрестные запросы»** предназначены для группирования данных и представления их в компактном виде. Запрос можно создать самостоятельно или воспользоваться **Мастером запросов**.

Создайте запрос на выборку с именем **«Номера телефонов»**.

1. Выберите закладку **«Запросы»**, если находитесь в другом диалоговом окне.
2. Щелкните мышкой по кнопке **«Создать»**.
3. В появившемся диалоговом окне выберите **«Конструктор»**, щелкните по кнопке **«ОК»**.

4. Добавьте нужные таблицы («*Личные данные*» и «*Список*»), выбирая их и щелкая по кнопке «*Добавить*». Так как таблицы связаны, то между ними появится линия, обозначенная цифрами 1 («*один-к-одному*»).

5. Закончите выбор, щелкнув по кнопке «*Заккрыть*». Появляется возможность выбора полей из разных таблиц.

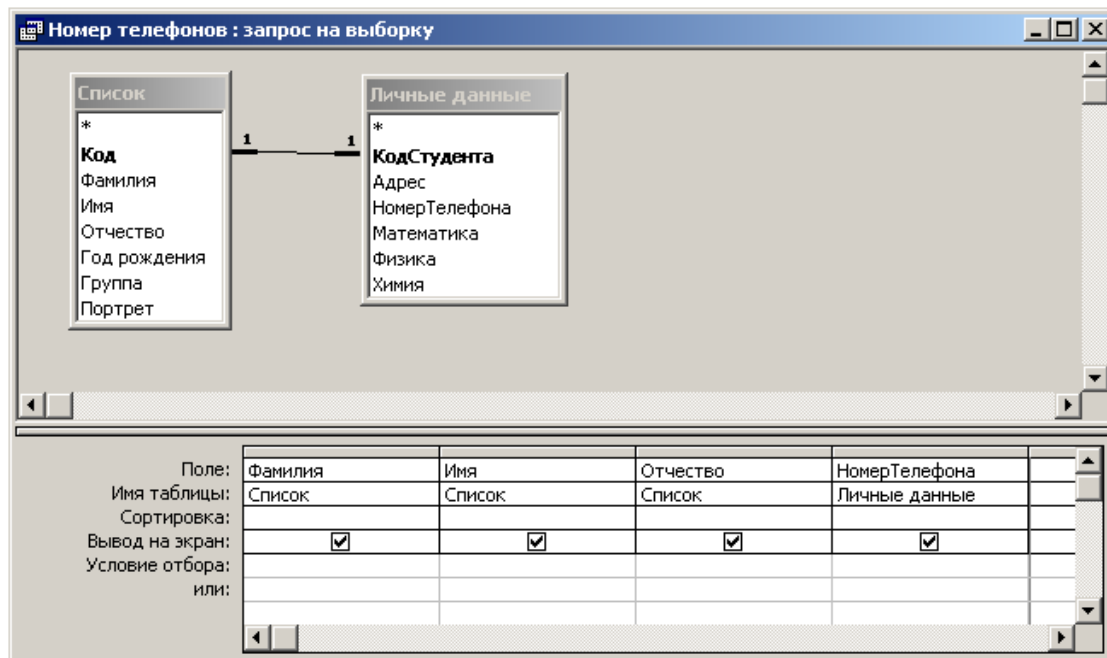



Рис. 5.46. Запрос в режиме Конструктора

6. Выберите поля «*Фамилия*», «*Имя*» и «*Отчество*» из таблицы «*Список*» и «*НомерТелефона*» – из таблицы «*Личные данные*». Для этого достаточно сделать двойной щелчок мышкой по имени поля. Второй вариант – перетащить мышкой название поля в клетки запроса.

7. Сохраните запрос, щелкнув по кнопке «*Сохранить*». Введите имя запроса «*Номера телефонов*» и щелкните по кнопке «*ОК*» (рис. 5.46).

8. Щелкните по кнопке  для представления запроса. Это самый простой вид запроса на выборку. В результате вы получаете новую таблицу с другим набором полей. Перейдите в режим «*Конструктор*».

9. Условие отбора можно включать аналогично включению фильтра. Например, телефонная книга для всех учащихся, фамилии которых начинаются на букву «*В*», может быть получена с помощью включения условия *Like «В\*»*. Получите этот запрос самостоятельно.

10. Измените имя запроса, выбрав в меню пункт «*Файл > Сохранить как/Экспорт*».

11. В появившемся диалоговом окне наберите новое имя запроса: «*Выборка по В*». Теперь в меню базы данных в окне «*Запросы*» будет показано два запроса.

12. Самостоятельное задание:

- 1) Составьте запрос на адреса только девушек, имя которых «*Анна*». Сохраните запрос с именем «*Анна*».
- 2) Составьте запрос на телефоны учащихся, отчество которых начинается на букву «*А*». Сохраните запрос с именем «*Выборка по А*».

### Задание 3

#### Создание запроса с использованием логических операций

Составьте запрос с использованием логических операций в условии отбора.

Предположим, что вам нужно составить ведомость для выплаты стипендии всем студентам, которые учатся без троек. Для этого нужно выбрать записи, в которых оценки по предметам **4 ИЛИ 5**.

1. Выберите закладку «*Запрос*», если находитесь в другом диалоговом окне.
2. Щелкните мышкой по кнопке «*Создать*».
3. В появившемся диалоговом окне выберите «*Конструктор*», щелкните по кнопке «*ОК*».
4. Добавьте нужные таблицы («*Личные данные*» и «*Список*»), выбирая их и щелкая по кнопке «*Добавить*». Так как таблицы связаны, между ними на экране появляется линия «*один-к-одному*». Если ее нет, таблицы нужно связать,
5. Закончите выбор, щелкнув по кнопке «*Заккрыть*». Появляется возможность выбора полей из разных таблиц.
6. Выберите поля «*Фамилия*», «*Имя*» и «*Отчество*» из таблицы «*Список*» и поля «*Математика*», «*Физика*», «*Химия*» – из таблицы «*Личные данные*». Для этого достаточно сделать двойной щелчок мышкой по имени поля. Второй вариант – перетащить мышкой название поля в клетки запроса.
7. В строке «*Условие отбора*» под полями «*Математика*», «*Физика*», «*Химия*» поставьте «*4 Or 5*» (рис. 5.47).

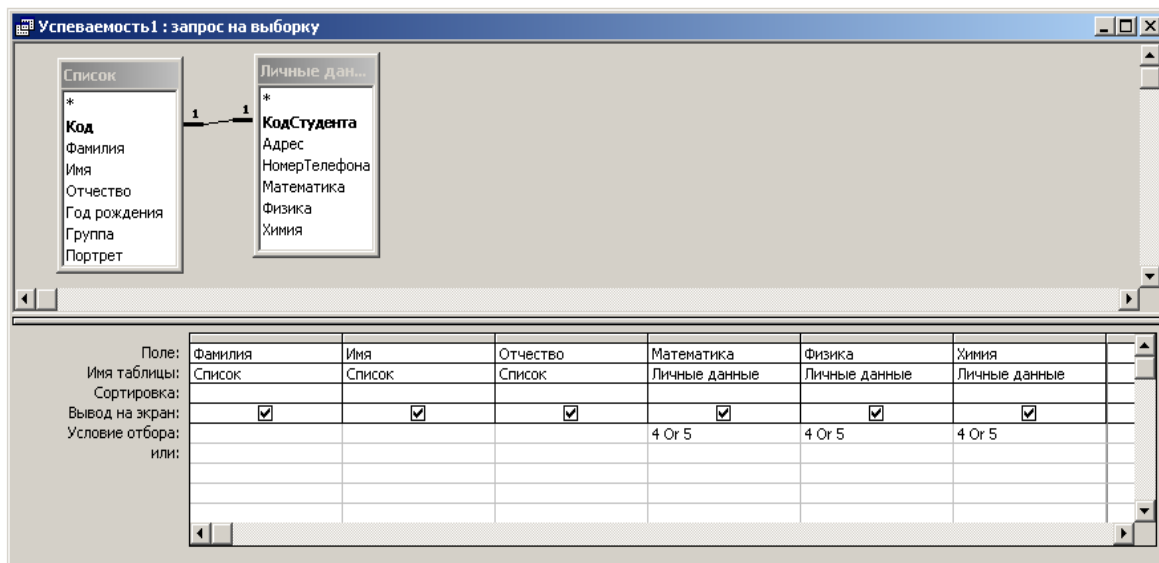




Рис. 5.47. Запрос в режиме конструктора

8. Щелкните по кнопке  представления запроса.
9. Сохраните запрос с именем «*Успеваемость1*», щелкнув по кнопке «*Сохранить*». Теперь в меню базы данных в окне «*Запросы*» будет показано три запроса.
  - 1) Самостоятельное задание: составьте запрос на студентов группы **1МС1**, у которых оценка по *математике* 4 или 5; сохраните запрос с именем «*Успеваемость2*».
10. Составьте запрос на студентов групп **1ТМ1** и **1МС1**, которые имеют оценку по курсу «*Физика*» и «*Химия*» 4 или 5; сохраните запрос с именем «*Успеваемость3*».



**Задание 4****Создание запроса на выборку**

Составьте запрос на выборку всех записей, кроме тех, в которых указана фамилия **Баранова** с использованием **Построителя выражений**.

1. Выберите закладку «**Запросы**», если находитесь в другом окне.
2. Выделите запрос «**Номера телефонов**».
3. Щелкните по кнопке «**Конструктор**».
4. Удалите поле «**НомерТелефона**».
5. Добавьте поле «**Адрес**».
6. Сохраните запрос с именем «**Адрес**», выполнив команду «**Сохранить как/экспорт...**».
7. Поставьте курсор в ячейку «**Условие отбора**» в столбце «**Фамилия**».
8. Удалите надпись в этой ячейке.
9. Щелкните по кнопке  («**Построить**»). Появится окно, в котором можно строить сложные запросы.
10. Щелкните по кнопке «**Not**», это слово появится в верхнем поле. Фамилию «**Баранова**» в кавычках наберите вручную.
11. Щелкните по кнопке «**ОК**». В поле «**Условие отбора**» появится данное выражение.
12. Щелкните по кнопке для представления запроса.
13. Закройте запрос, сохранив его с именем «**не\_Баранова**», выполнив команду «**Файл > Сохранить как/экспорт ...**».

**Задание 5****Создание вычисляемых полей**

Создайте вычисляемые поля.

1. Выберите закладку «**Запросы**», если находитесь в другом окне. Щелкните по кнопке «**Создать**».
2. В появившемся диалоговом окне выберите «**Конструктор**», щелкнув по кнопке «**ОК**».
3. Добавьте нужные таблицы («**Список**» и «**Личные данные**»), выбирая их и щелкая по кнопке «**Добавить**».
4. Завершите выбор, щелкнув по кнопке «**Закреть**».
5. Выберите поля «**Фамилия**» и «**Имя**» из таблицы «**Список**» и поля «**Физика**» и «**Химия**» – из таблицы «**Личные данные**».
6. Поставьте курсор на клетку правее «**Химия**» (на линии «**Поле**»).
7. Щелкните по кнопке  («**Построить**»).
8. В появившемся окне напечатайте вручную выражение, представленное на рис. 5.48, и щелкните по кнопке «**ОК**».
9. Это выражение подставится в новое поле. Нажмите клавишу «**Enter**».
10. Сохраните запрос с именем «**Среднее**».
11. Щелкните по кнопке  для представления запроса. Новое поле будет иметь имя «**Среднее**».
12. Закройте запрос.

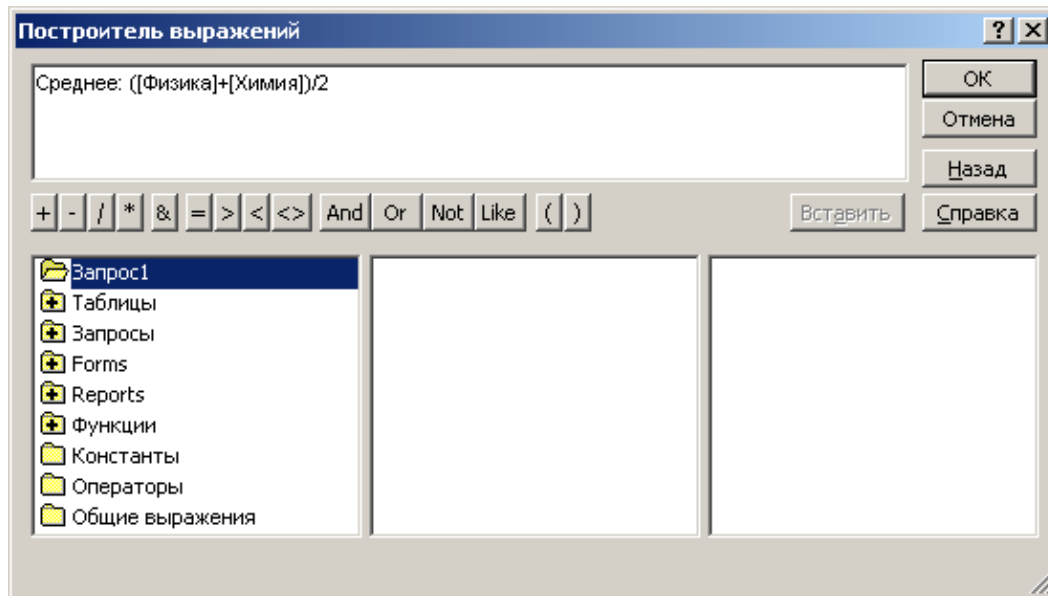


Рис. 5.48. Построитель выражений

**Задание 7****Самостоятельное задание**

1. Создайте новую таблицу с названием «*Новая группа*» и текстовыми полями «*Фамилия*», «*Имя*», «*Отчество*» и текстовым полем «*Группа*» с полем подстановок из таблицы «*Группы*». Для этого:
  2. Задайте ключевое поле «*Код*».
  3. Заполните значениями: Сидорова Анна Ивановна, Петрова Инна Сергеевна, Сергеева Ирина Петровна, Куликова Ольга Дмитриевна. Номер группы 1МС1.
  4. Заполните значениями: Алексеев Иван Петрович, Соловьева Ольга Петрова. Номер группы 0МС1.
  5. Закройте таблицу, предварительно сохранив ее.

**Задание 8****Запрос на обновление**

Постройте запрос на обновление.

Группу *1МС1* переименовали в *1СС1*. Нужно исправить значения поля «*Группа*» в таблице «*Группа*».

1. Выберите закладку «*Запрос*», если находитесь в другом окне.
2. Щелкните мышкой по кнопке «*Создать*».
3. В появившемся диалоговом окне выберите «*Конструктор*», щелкните по кнопке «*ОК*».
4. Добавьте таблицу «*Группа*», выбрав ее из списка и щелкнув по кнопке «*Добавить*».
5. Закончите выбор, щелкнув по кнопке «*Закреть*».
6. Выберите поле «*Группа*» из таблицы «*Группа*», щелкнув по нему мышкой два раза.
7. Щелкните по стрелке рядом с кнопкой «*Тип запроса*» на панели инструментов и выберите команду «*Обновление*».
8. Введите условие замены: в поле «*Группа*» заменить все группы *1МС1* на слово *1СС1*.
9. Выполните запрос.
10. Подтвердите обновление записей.
11. Закройте запрос, сохранив его с именем *1СС1*.
12. Выберите вкладку «*Таблицы*».
13. Откройте таблицу «*Список*». Просмотрите результат применения запроса.

**Задание 9****Запрос на добавление**

Постройте запрос на добавление. Записи из таблицы «*Новая группа*» нужно добавить в таблицу «*Список*».

1. Выберите закладку «*Запрос*», если находитесь в другом окне.
2. Щелкните мышкой по кнопке «*Создать*».
3. В появившемся диалоговом окне выберите режим «*Конструктор*», щелкните по кнопке «*ОК*».
4. Добавьте таблицу «*Новая группа*», выбрав ее и щелкнув по кнопке «*Добавить*».
5. Закончите выбор, щелкнув по кнопке «*Заккрыть*».
6. Выберите те поля, которые нужно добавить («*Фамилия*», «*Имя*», «*Отчество*», «*Группа*») согласно рис. 5.49.

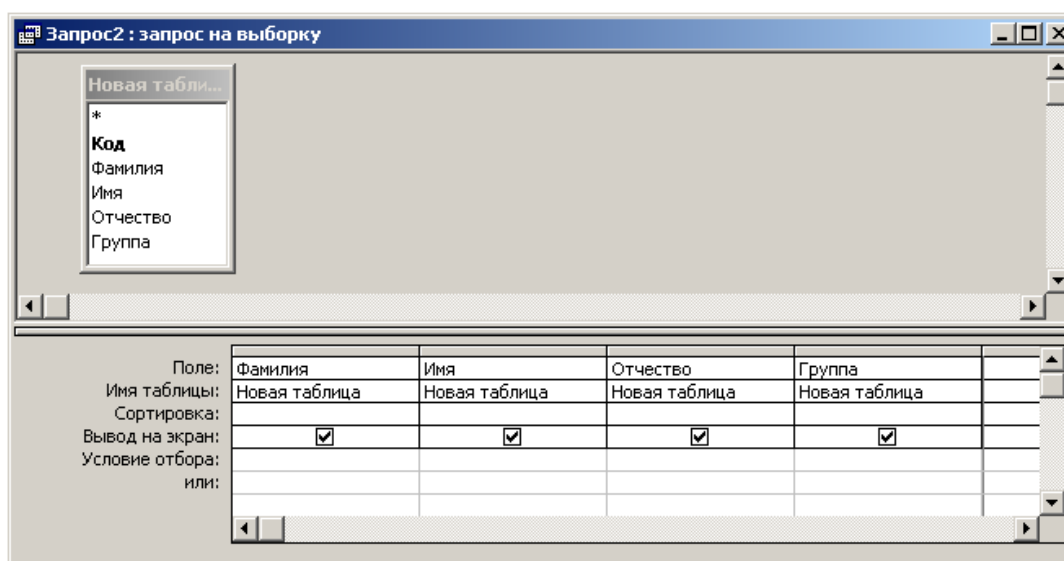




Рис. 5.49. Запрос на выборку

7. Щелкните по стрелке рядом с кнопкой  («*Тип запроса*») на панели инструментов и выполните команду «*Добавление*».

8. В появившемся диалоговом окне выберите имя таблицы «*Список*», в которую будут добавляться данные. Эта таблица может быть из текущей базы данных или из любой другой. В данном случае у нас одна база данных, поэтому щелкните по кнопке «*ОК*».

9. Щелкните по кнопке  для выполнения запроса. Подтвердите выполнение запроса.



10. Закройте запрос, сохранив его с именем «*Добавление*».

11. Перейдите на закладку «*Таблицы*». Откройте таблицу «*Список*» и убедитесь, что данные записи были добавлены.

**Задание 10****Запрос на удаление**

Постройте запрос на удаление. Нужно удалить записи тех учеников, которые окончили университет (например, тех, у которых группа была 0ТС1).

1. Выберите закладку «*Запрос*», если находитесь в другом окне.
2. Щелкните мышкой по кнопке «*Создать*».


3. В появившемся диалоговом окне выберите режим **«Конструктор»**, щелкните по кнопке **«ОК»**.
4. Добавьте таблицу **«Список»**, выбрав ее из списка и щелкнув по кнопке **«Добавить»**.
5. Закончите выбор, щелкнув по кнопке **«Заккрыть»**.
6. Щелкните по стрелке рядом с кнопкой **«Тип запроса»** на панели инструментов и выполните команду **«Удаление»**.
7. Выберите поле **«Группа»** из таблицы **«Список»**, щелкнув по нему мышкой два раза.
8. Введите условие отбора **0ТС1**.
9. Для предварительного просмотра удаляемых записей щелкните по кнопке  (**«Вид»**) на панели инструментов. Для возврата в режим **«Конструктор»** снова щелкните по этой кнопке на панели инструментов. Любые изменения следует вносить в режиме **«Конструктор»**.
10. Щелкните по кнопке  для выполнения запроса. Появится сообщение, что процесс удаления будет необратим. Подтвердите удаление записей.
11. Закройте запрос, сохранив его с именем **«Удаление»**.
12. Перейдите на закладку **«Таблицы»**. Откройте таблицу **«Список»** и убедитесь, что данные записи были удалены.

### Задание 11

#### Запрос на создание таблицы

Бывают ситуации, когда из старых таблиц нужно составить новые с другим набором полей. Предположим, что мы хотим создать таблицы успеваемости для учеников разных групп, поместив поля **«Фамилия»**, **«Имя»**, **«Отчество»** из таблицы **«Список»** и поля **«Математика»**, **«Физика»** и **«Химия»** – из таблицы **«Личные данные»**.

Выполните запрос на создание таблицы.

1. Выберите закладку **«Запрос»**, если находитесь в другом окне.
2. Щелкните мышкой по кнопке **«Создать»**.
3. В появившемся диалоговом окне выберите режим **«Конструктор»**, щелкнув по кнопке **«ОК»**.
4. Добавьте таблицы **«Список»** и **«Личные данные»**, выбрав их и щелкнув по кнопке **«Добавить»**.
5. Закончите выбор, щелкнув по кнопке **«Заккрыть»**.
6. Щелкните по стрелке рядом с кнопкой **«Тип запроса»** на панели инструментов и выполните команду **«Создание таблицы ...»**.
7. Напечатайте имя таблицы **«Успеваемость»** и щелкните по кнопке **«ОК»**.
8. Выберите поля **«Фамилия»**, **«Имя»**, **«Отчество»**, **«Группа»** из таблицы **«Список»** и поля **«Математика»**, **«Физика»** и **«Химия»** – из таблицы **«Личные данные»**.
9. Щелкните по кнопке  для выполнения запроса. Подтвердите выполнение запроса. Закройте запрос, сохранив его с именем **«Новая таблица»**.
10. Перейдите на закладку **«Таблицы»**. Откройте таблицу **«Успеваемость»** и убедитесь, что записи были добавлены. Причем добавлено только 10 записей, так как в таблицу **«Личные данные»** дополнительные записи не вошли.

## Задание 12

### Создание перекрестного запроса

Создайте перекрестный запрос.

Допустим, вам нужно подсчитать для экзаменационной ведомости, сколько в группе, занимающейся изучением *математики*, получено «троек», «четверок» и «пятерок». Для подобных целей используется перекрестный запрос.

1. Выберите закладку «*Запрос*», если находитесь в другом окне.
2. Щелкните мышкой по кнопке «*Создать*».
3. В появившемся диалоговом окне выберите «*Перекрестный запрос*», щелкнув по кнопке «*ОК*».
4. В окне «*Создание перекрестных запросов*» выделите таблицу «*Успеваемость*» и щелкните по кнопке «*Далее*».
5. Выберите поле, значения которого будут использоваться в качестве заголовков строк, например, «*Учебная группа*». Щелкните по кнопке «*Далее*».
6. Выберите поле, значения которого будут использоваться в качестве заголовков столбцов, например «*Математика*». Щелкните по кнопке «*Далее*».
7. Выберите функцию, по которой будут вычисляться значения ячеек на пересечении столбцов и строк (в данном случае «*Count*» - количество). Здесь вы можете добавить итоговое поле для каждой строки (в данном случае это поле покажет общее количество оценок по каждой группе). Щелкните по кнопке «*Далее*».
8. Задайте имя запроса «*Математика*». Щелкните по кнопке «*Готово*».
9. Самостоятельное задание
  - 1) Составьте аналогичные запросы для оценок, полученных группой по изучению *физики и химии*.

## Задание 13

### Завершение работы программы

Завершите работу с *Access*.

1. Выполните команду «*Файл > Выход*».
2. Если вы производили редактирование в базе данных, появится вопрос о сохранении изменений. Ответьте на него утвердительно.

## 5.7. СОЗДАНИЕ ОТЧЕТОВ

### Цели работы

- 1) научиться создавать отчеты.
- 2) закрепить навыки создания отчетов с помощью *Конструктора*;
- 3) научиться создавать отчеты с помощью *Мастера отчетов*;
- 4) освоить основные приемы изготовления надписей на конвертах и наклейках.

### Задание 1

#### Открытие базы данных

Откройте учебную базу данных, созданную ранее.

1. Запустите *Access*.
2. Выберите пункт «*Открыть базу данных*», выделите из списка баз данных, расположенного ниже переключателя, имя вашей базы и щелкните по кнопке «*ОК*». На экране появится окно с основными элементами базы данных.


## Задание 2

### Создание отчета

Отчет – это гибкое и эффективное средство для организации данных при выводе на печать и, вместе с тем, это способ вывода данных из базы на печать в том виде, в котором требуется пользователю, например, в виде справок об обучении, экзаменационных ведомостей, таблиц, объединенных каким-либо признаком, и др.

С помощью отчета можно расположить информацию на листе в удобном для пользователя виде с различным оформлением. Можно разработать отчет самостоятельно с помощью *Конструктора*, использовать готовые варианты оформления (*автоотчеты*) или создать отчет с помощью *Мастера*.


Создайте автоотчет в столбец на основании запроса «*Адрес*».

1. Откройте закладку «*Отчеты*», если находитесь в другом окне.
2. Щелкните по кнопке «*Создать*».
3. В появившемся диалоговом окне «*Новый отчет*» выберите «*Автоотчет: в столбце*» и запрос «*Адрес*».
4. Щелкните по кнопке «*ОК*». Появится страница просмотра отчета.
5. Сохраните отчет с именем «*Адрес*». Закройте отчет. Этот отчет составлен на основании запроса. При изменении запроса «*Адрес*» изменится и отчет. Это дает возможность, например, распечатать адрес только одного ученика.
6. Откройте закладку «*Запросы*», если находитесь в другом окне. Выделите запрос «*Адрес*».
7. Откройте запрос в режиме «*Конструктор*»: щелкните по кнопке «*Конструктор*».
8. Введите условие отбора фамилии – *Баранова*.
9. Выполните запрос, щелкнув по кнопке . Сохраните его и закройте.
10. Перейдите на закладку «*Отчеты*».
11. Откройте отчет «*Адрес*». В списке вывода будут находиться данные только одного человека.
12. Распечатайте данный отчет.

## Задание 3

### Создание ленточного автоотчета

Создайте ленточный автоотчет на основании запроса «*Номера телефонов*».


1. Щелкните по кнопке «*Создать*».
2. В появившемся диалоговом окне «*Новый отчет*» выберите «*Автоотчет: ленточный*» и запрос «*Номера телефонов*». Щелкните по кнопке «*ОК*». Появится страница просмотра отчета. Закройте страницу просмотра, щелкнув по кнопке «*Заккрыть*».
3. Сохраните отчет с именем «*Номера телефонов*». Закройте отчет. Для подобных отчетов удобно использовать «*Запрос с параметром*». Создадим запрос с параметром для запроса «*Номера телефонов*».
4. Откройте закладку «*Запросы*», если находитесь в другом окне.
5. Выделите запрос «*Номера телефонов*».
6. Откройте его в режиме «*Конструктор*»: щелкните по кнопке «*Конструктор*».
7. Напечатайте в качестве условия отбора в квадратных скобках фразу: [*Введите фамилию*]. Эти слова будут появляться каждый раз при выполнении запроса.
8. Выполните запрос, щелкнув по кнопке . Появится диалоговое окно «*Введите значение*» параметра с просьбой ввести фамилию человека, для которого вы хотите получить информацию.

9. Введите фамилию **Корнилова** и щелкните по кнопке **«ОК»**.
10. Сохраните запрос и закройте его.
11. Перейдите на закладку **«Отчеты»**.
12. Откройте отчет **«Номера телефонов»**. Появится точно такой же вопрос о фамилии. Введите любую фамилию, например **Корнилова**. В списке вывода будут данные только одного человека.

#### Задание 4

##### Изменения в отчетах

Внесите изменения в готовые отчеты.

1. Перейдите в режим **«Конструктор»**. Все изменения в отчет можно вводить только в этом режиме. В окне **«Конструктор отчетов»** находится пять областей:
  - 1) **«Заголовок отчета»** – все, что находится в этой области, выводится только один раз в начале отчета;
  - 2) **«Верхний колонтитул»** – все, что находится в этой области, выводится в верхней части каждой страницы;
  - 3) **«Область данных»** – содержит собственно записи;
  - 4) **«Нижний колонтитул»** – все, что находится в этой области, выводится в нижней части каждой страницы;
  - 5) **«Примечание отчета»** – все, что находится в этой области, выводится только один раз в конце отчета.
2. В окне **«Конструктор отчетов»** в заголовке отчета находится надпись **«Номера телефонов»**, в верхнем колонтитуле – заголовки столбцов вывода, в области данных – записи, в нижнем колонтитуле – дата текущего дня.
3. Исправьте заголовок отчета на **«Номер телефона учащегося»**. Смените цвет букв, их размер и шрифт.
4. Перейдите в режим предварительного просмотра, щелкнув по кнопке . Введите фамилию из списка учащихся и посмотрите, что получилось.
5. Закройте отчет, предварительно сохранив его.

#### Задание 5


##### Создание отчета 1

Создайте отчет **«Справка»** с помощью **«Конструктора»**.

Рассмотрим ситуацию, когда стандартный отчет нас не устраивает. Например, вы хотите сконструировать стандартную справку об обучении и выдавать ее по запросу. Сначала следует создать запрос с параметром **«Справка»**, в котором будут только интересные вас записи, затем следует приступить к созданию отчета.

1. Откройте закладку **«Отчеты»**, если находитесь в другом окне.
2. Щелкните по кнопке **«Создать»**.
3. В появившемся диалоговом окне **«Новый отчет»** выберите режим **«Конструктор»** и запрос **«Справка»**.
4. Щелкните по кнопке **«ОК»**. Появится **«Конструктор»** для создания отчетов и панель с вспомогательными кнопками. Наличие или отсутствие заголовка и примечания отчета выбирается командой **«Вид > Колонтитулы и Вид > Заголовок/Примечание отчета»**. При включении соответствующего переключателя появляется соответствующий раздел **«Конструктора»**. Если отсутствует панель элементов, щелкните по кнопке
5. В поле верхнего колонтитула поместите надпись:

Университет  
СПРАВКА

6. Для этого щелкните по кнопке  («Надпись»), расположенной на дополнительной панели. Укажите мышкой место начала надписи и введите текст с клавиатуры. В конце ввода слов «*Университет*» нажмите одновременно комбинацию клавиш «*Shift + Enter*».

7. В конце ввода «*СПРАВКА*» нажмите клавишу «*Enter*». Сделайте надпись размером 18. Выполните команду «*Формат > Размер > по размеру данных*». Расположите надпись симметрично по центру страницы.

8. В области данных напечатайте три строки:

Дана настоящая в том, что  
является студентом (кой)  
группы

9. Щелкните по кнопке  («Список полей»).

10. В открывшемся окне выберите все поля, выделяя их мышкой при нажатой клавише «*Ctrl*», и перетащите их мышкой в область данных.

11. Удалите названия полей. Оставьте только содержимое записей. Сделайте всю запись шрифтом 12 и расположите ее в удобном для вас виде.

12. Выделите все элементы области данных при нажатой клавише «*Shift*».

13. Выровняйте элементы по левому и по нижнему краю, выполнив соответственно команду «*Формат > Выровнять > По левому краю и Формат > Выровнять > По нижнему краю*». Уберите пустое место в области данных, передвинув границу этой области.

14. Выберите параметры страницы. Поля: верхнее и нижнее – по 20 мм, а правое и левое – по 10 мм. Ориентация: альбомная. В разделе «*Нижний колонтитул*» напечатайте подпись: «*Декан института*».

15. Сохраните макет отчета с именем «*Справка*». Перейдите в режим предварительного просмотра.

16. Введите из списка фамилию учащегося и посмотрите, что получилось.

17. Закройте отчет, предварительно сохранив его.

## Задание 6

### Создание отчета 2

Создайте с помощью «*Конструктора*» отчет «*Списки учеников*». Ставится задача вывести списки учащихся по группам. Для этого их нужно сгруппировать и отсортировать внутри группы по алфавиту. В качестве источника данных возьмем таблицу «*Список*».

1. Откройте закладку «*Отчеты*», если находитесь в другом окне.

2. Щелкните по кнопке «*Создать*».

3. В появившемся диалоговом окне «*Новый отчет*» выберите режим «*Конструктор*» и таблицу «*Список*» в качестве источника данных. Щелкните по кнопке «*ОК*».


4. Если отсутствует раздел «*Заголовок отчета*», добавьте его с помощью команды «*Вид > Заголовок/примечание отчета*».

5. Щелкните по кнопке  («*Сортировка и группировка*»).

6. В появившемся окне выберите поле «*Учебная группа для группировки*». В разделе «*Свойства группы*» в заголовке группы включите «*Да*». В результате будет создан заголовок группы.

7. Выберите второе поле для сортировки «*Фамилия*». У этого поля в заголовке группы значение «*Нет*»

8. Закройте это окно.


9. Напечатайте в разделе заголовка отчета: «**Список студентов института**». Для этого щелкните по кнопке  – «**Надпись**» на дополнительной панели. Укажите мышкой место начала надписи и введите текст с клавиатуры. В конце ввода нажмите клавишу «**Enter**». Надпись, приведенная в разделе заголовка отчета, будет единственной для всего отчета.

10. Сделайте надпись шрифтом 16. Выполните команду «**Формат > Размер > по размеру данных**». Измените размер окна, если надпись видна не полностью.

11. Напечатайте в разделе «**Верхний колонтитул**» надпись «**Список студентов**» шрифтом 14. Эта надпись будет появляться на каждой странице. Передвиньте надпись на правый край страницы.

12. Напечатайте в разделе «**Заголовок группы**» надпись «**Список студентов**», затем поместите поле «**Группа**» (удалите подпись поля), а потом – слово «**группы**».

13. Поместите поля «**Фамилия**», «**Имя**», «**Отчество**», «**Год рождения**» в раздел «**Область данных**». Уберите подписи полей. Сделайте размер 12. Выполните команду «**Формат > Размер > по размеру данных**».

14. В разделе «**Нижний колонтитул**» поместите номера страниц. Для этого добавьте в область данных пустое поле, щелкнув по кнопке  и перетащив ее мышкой в нужное место.

15. Удалите подпись поля. Выделите пустое поле.

16. Включите кнопку  («**Свойства**»).

17. Наберите в строке «**Данные**» значение:  
= [Page].

18. Закройте текущее окно. Номера страниц будут появляться в конце каждой страницы.

19. Ниже надписи в разделе «**Заголовок группы**» проведите линию. Для этого воспользуйтесь кнопкой панели элементов «**Линия**».


20. Отделите линиями все заголовки. Выбрав режим предварительного просмотра документа, вы увидите его таким, каким он будет напечатан на бумаге. Линии сетки можно убрать (выберите пункт меню «**Вид**» и отключите галочку у пункта меню «**Сетка**»).

## Задание 7

### Нумерация записей

Пронумеруйте записи в области данных отчета.

1. Установите для отчета режим «**Конструктор**».

2. Добавьте в область данных пустое поле, щелкнув по кнопке  и перетащив ее мышкой в область данных в положение перед полем «**Фамилия**».

3. Удалите надпись пустого поля.

4. Выделите поле и щелкните по кнопке  – **Свойства** на панели инструментов.

5. Введите в ячейку «**Данные**» выражение:  
=1.

6. В ячейке свойства «**Сумма с накоплением**» выберите значение «**Для всего**». Во время предварительного просмотра или вывода отчета на печать номер каждой следующей записи будет увеличиваться на 1, т.е. будет 1,2,3....


7. Закройте окно со свойствами полей.

8. Перейдите в режим предварительного просмотра. Просмотрите результат.

9. Закройте отчет, предварительно сохранив его под именем «**Список учеников**».

**Задание 8****Подсчет записей**

Подсчитайте количество записей в отчете.

1. Откройте отчет Список учеников в режиме *Конструктора*.
2. Добавьте область примечания отчета, если ее нет в *Конструкторе*. Для этого выполните команду «*Вид > Заголовок/Примечание отчета*».
3. Добавьте в область примечания пустое поле, щелкнув по кнопке **abl** и перетащив ее мышкой в область «*Примечание отчета*» в положение под полем «*Фамилия*».
4. Убедитесь, что нужное (пустое) поле выбрано, и щелкните по кнопке  – («*Свойства*») на панели инструментов.
5. Введите в ячейку «*Свойства-Данные*» вычисляемого поля выражение:  
=Count(\*)

Это выражение использует функцию «*Count*» для подсчета всех записей отчета, включая записи, имеющие пустые значения в некоторых полях.

6. Введите название: «*Общее количество записей*».
7. В режиме предварительного просмотра вы увидите подсчитанным общее количество записей.
8. Сохраните отчет под тем же именем. Закройте отчет.

**Задание 9****Создание отчета с помощью Мастера отчетов**

Создайте отчет с помощью *Мастера отчетов*.

1. Откройте закладку «*Отчеты*», если находитесь в другом окне.
2. Щелкните по кнопке «*Создать*».
3. В появившемся диалоговом окне «*Новый отчет*» выберите «*Мастер отчетов*» и таблицу «*Список*». Щелкните по кнопке «*ОК*».
4. В появившемся диалоговом окне создания отчета выберите поля «*Фамилия*», «*Имя*», «*Отчество*», «*Группа*».
5. Щелкните по кнопке «*Далее*». В результате появится новое диалоговое окно, где следует выбрать поля для группировки (объединения по признаку).
6. С помощью кнопок «*Уровень*» можно менять местами уровни группировки.
7. Щелкните по кнопке «*Далее*».
8. На этом шаге можно отсортировать данные.
9. Щелкните по кнопке «*Далее*».
10. Выберите вид макета для отчета ступенчатый и щелкните по кнопке «*Далее*».
11. Выберите стиль отчета: «*Выделенный*» и щелкните по кнопке «*Далее*».
12. Задайте имя отчета: «*Отчет1*» и щелкните по кнопке «*Готово*». Вы попадете в режим просмотра отчета. Возможно переключение в три режима: «*Конструктор*», «*Предварительный просмотр*» и «*Образец*».
13. Закройте отчет, согласившись с сохранением.
14. Самостоятельное задание.
  - 1) Составьте отчеты, выбирая разные макеты:
    - а) Отчет0 – блок;
    - б) Отчет2 – структура 1;
    - в) Отчет3 – структура 2;
    - г) Отчет4 – по левому краю 1;
    - д) Отчет5 – по левому краю 2;
  - 2) составьте отчеты, выбирая разные стили:

- а) Отчет6 – компактный;
- б) Отчет7 – простой;
- в) Отчет8 – строгий;
- г) Отчет9 – формальный;
- д) Отчет10 – черно-белый.

### **Задание 10**

#### **Создание почтовых наклеек**

Создайте почтовые наклейки.

1. Откройте закладку «**Отчеты**». Щелкните по кнопке «**Создать**».
2. В появившемся диалоговом окне «**Новый отчет**» выберите «**Почтовые наклейки**» и таблицу «**Список**». Щелкните по кнопке «**ОК**».
3. В следующем окне щелкните по кнопке «**Далее**».
4. В следующем окне вновь щелкните по кнопке «**Далее**».
5. В следующем окне создайте прототип наклейки, напечатав нужные слова и выбрав соответствующие поля. Щелкните по кнопке «**Далее**».
6. В следующем окне укажите поля для сортировки (например, «**Фамилия**» – по алфавиту). Щелкните по кнопке «**Далее**».
7. Введите имя отчета «**Наклейки**» и щелкните по кнопке «**Готово**».

### **Задание 11**

#### **Завершение работы с программой**

Завершите работу с программой *Access*.

1. Выполните команду «**Файл > Выход**».
2. Если вы производили редактирование в базе данных, появится вопрос о сохранении изменений. Ответьте на него положительно.

## КОНТРОЛЬНАЯ РАБОТА

### ЗАДАНИЕ

Необходимо разработать базу данных по предложенной теме. В базе данных необходимо наличие всех объектов, необходимых для решения задач работы с информацией данной предметной области.

### ВАРИАНТЫ

1. **Личная библиотека.** Картотека домашней библиотеки: выходные данные книги (авторы, название, издательство и так далее), раздел библиотеки (специальная литература, хобби, домашнее хозяйство, беллетристика и так далее), происхождение и наличие книги в данный момент, субъективная оценка книги. Выбор книг по произвольному запросу; инвентаризация библиотеки.

2. **Записная книжка.** Анкетные данные, адреса, телефоны, место работы или учебы, должность знакомых, коллег и родственников, характер знакомства, деловые качества и так далее. Автоматическое формирование поздравления с днем рождения (по текущей дате). Упорядочение по алфавиту и по дате последней корректировки. Поиск по произвольному шаблону.

3. **Справочник покупателя.** База торговых точек города: название, адрес и телефоны, специализация, форма собственности, время работы. Выбор магазинов по произвольному шаблону.

4. **Генеалогическое дерево.** Паспортные данные членов некоторого родового клана; ссылки на детей (или на родителей). Поиск всех потомков или всех предков для указанного лица.

5. **Склад.** База товаров, хранящихся на складе: наименование, единица измерения, цена единицы, количество, дата последнего завоза. Регистрация поступления товара (формирование приходной накладной) и отгрузки (расходная накладная). Вывод инвентарной ведомости.

6. **Справочник меломана.** База групп и исполнителей; база песен; база дисков с перечнем песен (в виде ссылок). Выбор всех песен заданной группы; всех дисков, где встречается заданная песня.

7. **Ежедневник.** База намечаемых мероприятий: дата, время и протяженность, место проведения. Автоматическое напоминание ближайшего дела: по текущей дате и времени; удаление вчерашних дел либо перенос на будущее. Анализ «накладок» – пересечений планируемых дел. Просмотр дел на завтра, послезавтра и так далее.

8. **Терминология.** База определений какой-либо науки: вводимый термин, его толкование (определение), ссылки на используемые термины. Возможность просмотра всей цепочки от заданного термина до первичных понятий.

9. **Шеф-повар.** База рецептов блюд: раскладка, рецепт приготовления. База продуктов на складе: наименование, цена, количество. Формирование меню на день (на заданное число персон); званый ужин. Проверка достаточности запасов; формирование расходной накладной на склад, корректировка запасов.

10. **Справочник абитуриента.** База вузов: наименование, адрес, перечень специальностей, конкурс прошлого года по каждой специальности (дневной, вечерней, заочной форм), размер оплаты при договорном обучении. Выбор по разным критериям: все

о данном вузе; все о данной специальности, поиск минимального конкурса по данной специальности или вообще.

11. **Справочник селекционера.** Наименование сорта какой-либо культуры, автор, родительские сорта, урожайность, характеристики плодов, морозоустойчивость, устойчивость к вредителям и болезням, наличие в том или ином селекционном фонде. Выбор сортов, обладающих заданными свойствами.

12. **Справочник владельца видеотеки.** База видеофильмов: название, студия, жанр, год выпуска, режиссер, исполнители главных ролей, краткое содержание, субъективная оценка фильма. Факт наличия фильма в видеотеке. Оформление выдачи и возврата кассеты.

13. **Справочник фаната.** База спортсменов: анкетные и антропологические данные, гражданство, происхождение, вид спорта, клуб или команда, данные о личном рекорде или победах и так далее. Выбор по произвольному признаку. Поиск рекордсмена в заданном виде спорта.

14. **Справочник радиолюбителя.** Базы паспортных данных транзисторов, диодов, тиристоров и так далее: марка, характеристики, предельно допустимые условия эксплуатации, цена, учет взаимозаменяемости и т. д. Подборка по заданным требованиям.

15. **Автосалон.** База новых и подержанных отечественных и иностранных автомобилей: марка, год выпуска, технические характеристики, особенности исполнения, техническое состояние, запрашиваемая цена. База покупателей: контактные координаты, требования к модели, техническим характеристикам и техническому состоянию, финансовые возможности. Автоматизация подбора вариантов для покупателя, формирование заявки для поставщиков и перегонщиков.

16. **Справочник туриста.** Турагентства и предлагаемые услуги: страна, город (или маршрут круиза), условия проживания и проезда, экскурсионное обслуживание, сервис принимающей стороны, стоимость путевки.

17. **Крылатые фразы.** Справочник пословиц, поговорок, афоризмов, каламбуров, других словесных курьезов. Классификация по авторам и источникам, поиск по темам и ключевым словам.

18. **Каталог запчастей автомобиля.** В автомобиле насчитывается несколько тысяч деталей; некоторые используются в разных марках. Таблицы: страна, фирма-изготовитель, марка автомобиля, агрегат, узел, деталь. Учет взаимозаменяемости. Пользователи: работники автосервиса, магазинов запчастей; поставщики-оптовики.

19. **Каталог радиодеталей (справочник радиомастера).** Модели бытовой аудио- и видеотехники; для каждой модели – каталог радиодеталей, использованных в ней (резисторов, конденсаторов, катушек индуктивности, чипов и т. д.). Многие детали используются в разных моделях; некоторые детали односторонне или двусторонне заменяемы. Учет наличия деталей на складе, поиск подходящих деталей для ремонта.

20. **Справочник нумизмата.** Монеты: страна, номинал, год выпуска, количество выпущенных монет, особенности. Коллекционеры: страна, имя, контактные координаты, наличие редких монет в коллекции. Собственная коллекция.

21. **Справочник филателиста.** Марки: страна, нарицательная стоимость, год выпуска, тираж, особенности. Филателисты: страна, имя, контактные координаты, наличие редких марок в коллекции. Собственная коллекция.

22. **Справочник любителя живописи.** Художники с анкетными данными и стилями. Картины со ссылкой на художников, датой создания, жанром. Коллекционеры и

---

музеи: наличие оригиналов картин и копий. Аукционы и комиссионки: дата проведения, список выставленных шедевров и цены на них. Собственная коллекция.

23. **Телепрограмма.** Программа телепередач нескольких телекомпаний (на неделю по дням, часам). Разные жанры у телепередач: новости, спорт, художественные фильмы, сериалы и т. д. Выбор совокупной программы по определенному запросу (вкусу). Программирование видеомагнитофона при временных «накладках» передач.

24. **Справочник астронома.** Для каждой из зарегистрированных звезд известны: название, созвездие, видимая звездная величина, расстояние, координаты на небосклоне. Поиск звезд заданного созвездия, самых ярких звезд, видимых звезд и созвездий в заданной точке земного шара в заданное время.

25. **Справочник географа.** Города (географические координаты, численность населения), регионы (области, провинции, штаты и т. д. – принадлежность стране, столица, численность населения), страны (площадь, численность населения, форма государственного правления, столица), материки. Поиск городов-«тезок», подсчет доли городского населения в странах и регионах, населенность материков и т. д.

26. **Справочник гидролога.** Реки мира: протяженность (от истока до впадения), куда впадает (в другую реку, море или озеро), годовой сток ( $\text{км}^3$ ), площадь бассейна (без крупных притоков). Вычисление годового стока и площади бассейна для заданной крупной реки (вместе с ее притоками), моря или озера.

## ЗАКЛЮЧЕНИЕ

Современное общество характеризуется небывалым ростом объема информационных потоков и в сфере экономики, и в социальной сфере. Особенно это касается отраслей промышленности, торговли, финансово-кредитной деятельности. В промышленности такой рост обусловлен увеличением объема производства, усложнением выпускаемой продукции, используемых материалов, технологического оборудования, расширением производства, внешних и внутренних связей экономических объектов. Рыночные отношения предъявляют повышенные требования к своевременности, достоверности и полноте информации, без чего немыслима эффективная маркетинговая, финансово-кредитная, инвестиционная деятельность.

К известным видам ресурсов (материальным, трудовым, энергетическим, финансовым) прибавился новый – информационный. Только на основе своевременного пополнения, накопления, переработки информационного ресурса, т. е. владения достоверной информацией, возможны рациональное управление любой сферой человеческой деятельности, принятие правильных решений. Особенно актуально это для сферы экономики. Применение современных компьютеров дает возможность переложить трудоемкие операции на автоматические или автоматизированные устройства, которые могут работать со скоростью, превышающей скорость обработки информации человеком в миллионы раз.

Использование компьютера приводит к коренной перестройке технологии производства практически во всех отраслях промышленности, коммерческой и финансово-кредитной деятельности и, как следствие, – к повышению производительности и улучшению условий труда людей. Именно поэтому современный специалист должен владеть теоретическими знаниями в области информатики и практическими навыками использования вычислительной техники, техники связи и других средств управления.

---

## ЛИТЕРАТУРА

1. **Острейковский, В.А.** Информатика. Теория и практика: учеб. пособие / В. А. Острейковский, И. В. Полякова. – М.: Оникс, 2008. – 608 с.
2. **Першиков, В.И.** Толковый словарь по информатике / В. И. Першиков, В. М. Савинков. – М.: Финансы и статистика, 1991.
3. Словарь по кибернетике / под ред. В.С. Михалевича. – 2-е изд. – Киев: Гл. ред. УСЭ, 1989.
4. Википедия. Свободная энциклопедия [Электронный ресурс]. Wikipedia® – зарегистрированная торговая марка Wikimedia Foundation, Inc., некоммерческой организации. – Режим доступа: <http://ru.wikipedia.org> свободный. – Загл. с экрана. Яз. рус., англ.

---

<b>ОГЛАВЛЕНИЕ</b>	
<b>ВВЕДЕНИЕ .....</b>	<b>3</b>
<b>1. ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ.....</b>	<b>4</b>
1.1. ЧЕЛОВЕК И ИНФОРМАЦИЯ .....	4
1.1.1. ТЕРМИНОЛОГИЯ ИНФОРМАТИКИ.....	4
1.1.2. ОБЪЕКТ И ПРЕДМЕТ ИНФОРМАТИКИ .....	6
1.1.3. ВИДЫ, ОСОБЕННОСТИ И ФОРМЫ ИНФОРМАЦИОННОГО РЕСУРСА.....	8
1.1.4. КРАТКАЯ ИСТОРИЯ РАЗВИТИЯ ИНФОРМАТИКИ .....	9
1.2. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ .....	10
1.2.1. ЭТАПЫ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА В ИНФОРМАЦИОННЫХ СИСТЕМАХ.....	10
1.2.2. ИНФОРМАЦИОННЫЕ ОСНОВЫ ПРОЦЕССОВ УПРАВЛЕНИЯ .....	13
1.2.3. ИНФОРМАТИЗАЦИЯ ОБЩЕСТВА .....	14
<b>2. СИСТЕМЫ СЧИСЛЕНИЯ И ОСНОВЫ ЛОГИКИ.....</b>	<b>18</b>
2.1. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ. КОЛИЧЕСТВО И ЕДИНИЦЫ ИЗМЕРЕНИЯ ИНФОРМАЦИИ .....	18
2.1.1. ЯЗЫК КАК СПОСОБ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ.....	18
2.1.2. ФОРМЫ ПРЕДСТАВЛЕНИЯ И ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ.....	19
2.1.3. КОЛИЧЕСТВО И ЕДИНИЦЫ ИЗМЕРЕНИЯ ИНФОРМАЦИИ.....	20
2.2. СИСТЕМЫ СЧИСЛЕНИЯ, ИСПОЛЬЗУЕМЫЕ В КОМПЬЮТЕРЕ.....	23
2.2.1. ПОНЯТИЕ СИСТЕМЫ СЧИСЛЕНИЯ.....	23
2.2.2. ПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ .....	23
2.2.3. ДВОИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ.....	25
2.2.4. ПРАВИЛА ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ В ДВОИЧНОЙ СИСТЕМЕ СЧИСЛЕНИЯ .....	25
2.2.5. ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ .....	26
2.2.6. ДРУГИЕ ПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ .....	27
2.3. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В ПАМЯТИ КОМПЬЮТЕРА .....	27
2.3.1. ПРЕДСТАВЛЕНИЕ ЦЕЛЫХ ЧИСЕЛ БЕЗ ЗНАКА И СО ЗНАКОМ.....	27
2.3.2. ИНДИКАТОРЫ ПЕРЕНОСА И ПЕРЕПОЛНЕНИЯ.....	29
2.3.3. ПРЕДСТАВЛЕНИЕ СИМВОЛЬНОЙ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ .....	30
2.3.4. ФОРМАТЫ ДАННЫХ.....	31
2.3.5. ПРЕДСТАВЛЕНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ .....	33
<b>3. КОМПЬЮТЕР.....</b>	<b>35</b>
3.1. ОСНОВНЫЕ УСТРОЙСТВА КОМПЬЮТЕРА.....	35
3.1.1. ОБЩАЯ СХЕМА ФУНКЦИОНИРОВАНИЯ КОМПЬЮТЕРА .....	35
3.1.2. ОСНОВНЫЕ БЛОКИ И УСТРОЙСТВА КОМПЬЮТЕРА .....	35
3.1.3. АРХИТЕКТУРА КОМПЬЮТЕРА .....	37
3.1.4. МАГИСТРАЛЬНО-МОДУЛЬНЫЙ ПРИНЦИП ПОСТРОЕНИЯ КОМПЬЮТЕРА.....	38
3.1.5. ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕР .....	41
3.2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА.....	41
3.3. ФАЙЛОВАЯ СИСТЕМА. РАБОТА С НОСИТЕЛЯМИ ИНФОРМАЦИИ .....	43
3.3.1. ФАЙЛОВАЯ СИСТЕМА.....	43
3.3.2. ГРАФИЧЕСКИЕ ПОЛЬЗОВАТЕЛЬСКИЕ ИНТЕРФЕЙСЫ .....	45
3.4. КОМПЬЮТЕРНЫЕ ВИРУСЫ И АНТИВИРУСНЫЕ ПРОГРАММЫ.....	46
3.4.1. КОМПЬЮТЕРНЫЕ ВИРУСЫ .....	46
3.4.2. АНТИВИРУСНЫЕ ПРОГРАММЫ И ЗАЩИТА ИНФОРМАЦИИ.....	46
3.5. ИСТОРИЯ И ПЕРСПЕКТИВЫ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ .....	49
3.5.1. ИСТОРИЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ.....	49
3.5.2. ПОКОЛЕНИЯ КОМПЬЮТЕРОВ .....	52
3.5.3. ПЕРСПЕКТИВЫ РАЗВИТИЯ ПК.....	55
<b>4. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ .....</b>	<b>59</b>
4.1. КЛАССИФИКАЦИЯ И ХАРАКТЕРИСТИКА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ .....	59
4.1.1. ПОНЯТИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ.....	59
4.1.2. СОДЕРЖАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ .....	60
4.2. ТЕХНОЛОГИЯ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ .....	62
4.2.1. ТЕКСТОВЫЙ РЕДАКТОР: НАЗНАЧЕНИЕ И ОСНОВНЫЕ ФУНКЦИИ .....	62
4.2.2. ИЗДАТЕЛЬСКИЕ СИСТЕМЫ .....	63

---

4.3.	ТЕХНОЛОГИЯ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ.....	64
4.3.1.	ВИДЫ ГРАФИКИ.....	64
4.3.2.	ГРАФИЧЕСКИЕ РЕДАКТОРЫ .....	64
4.4.	ТЕХНОЛОГИЯ ОБРАБОТКИ ЧИСЛОВОЙ ИНФОРМАЦИИ .....	65
4.5.	ТЕХНОЛОГИЯ ХРАНЕНИЯ, ПОИСКА И СОРТИРОВКИ ИНФОРМАЦИИ .....	66
4.5.1.	СПОСОБЫ ОРГАНИЗАЦИИ БАЗ ДАННЫХ.....	66
4.5.2.	СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ.....	67
4.5.3.	ТЕХНОЛОГИЯ ИСПОЛЬЗОВАНИЯ СУБД .....	68
4.6.	МУЛЬТИМЕДИЙНЫЕ ТЕХНОЛОГИИ .....	68
4.7.	КОМПЬЮТЕРНЫЕ КОММУНИКАЦИИ.....	69
4.7.1.	ПЕРЕДАЧА ИНФОРМАЦИИ. ЛИНИИ СВЯЗИ, ИХ ОСНОВНЫЕ КОМПОНЕНТЫ И ХАРАКТЕРИСТИКИ .....	69
4.7.2.	КОМПЬЮТЕРНЫЕ ТЕЛЕКОММУНИКАЦИИ: НАЗНАЧЕНИЕ, СТРУКТУРА, РЕСУРСЫ.....	70
4.7.3.	ЛОКАЛЬНЫЕ КОМПЬЮТЕРНЫЕ СЕТИ .....	73
4.7.4.	ГЛОБАЛЬНЫЕ КОМПЬЮТЕРНЫЕ СЕТИ .....	74
<b>5.</b>	<b>БАЗЫ ДАННЫХ .....</b>	<b>76</b>
5.1.	ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ.....	76
5.1.1.	УРОВНИ ДАННЫХ БД.....	76
5.1.2.	ЭТАПЫ ПРОЕКТИРОВАНИЯ БД.....	76
5.1.3.	ИНФОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ .....	77
5.2.	СОЗДАНИЕ БАЗЫ ДАННЫХ, СОСТОЯЩЕЙ ИЗ ОДНОЙ ТАБЛИЦЫ .....	83
5.3.	СОЗДАНИЕ БАЗЫ ДАННЫХ, СОСТОЯЩЕЙ ИЗ ДВУХ ТАБЛИЦ.....	90
5.4.	СОЗДАНИЕ БАЗЫ ДАННЫХ, СОСТОЯЩЕЙ ИЗ ТРЕХ ТАБЛИЦ.....	95
5.5.	ПРИМЕНЕНИЕ ФОРМ.....	102
5.6.	СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ЗАПРОСОВ .....	110
5.7.	СОЗДАНИЕ ОТЧЕТОВ .....	117
	<b>КОНТРОЛЬНАЯ РАБОТА .....</b>	<b>124</b>
	<b>ЗАКЛЮЧЕНИЕ.....</b>	<b>127</b>
	<b>ЛИТЕРАТУРА.....</b>	<b>128</b>

Учебное издание

**А.А. Просолович**

**КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ  
И ИНФОРМАТИКА**

**Учебное пособие**

Редактор Г.Н. Горькавая

Подписано в печать 21.05.12.

Формат 60 x 84 1/8. Бумага 80 г/м<sup>2</sup>. Отпечатано на ризографе.

Усл. печ. л. 16,1. Уч.-изд. л. 9,92. Тираж. Заказ.

Институт новых информационных технологий  
Федерального государственного бюджетного образовательного учреждения  
высшего профессионального образования  
«Комсомольский-на-Амуре государственный технический университет»