

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»

ИНФОРМАТИКА. БАЗОВЫЙ КУРС

Учебное пособие

Курган 2017

УДК 681.3 (075.8)

ББК 32.97 я7

И 74

Рецензенты

канд. физ.-мат. наук, зав. кафедрой математики и прикладной информатики Курганского филиала АТиСО **С.В. Косовских**;

д-р пед. наук, зав. кафедрой информационных технологий УрФУ **Л.И. Долинер**.

Печатается по решению методического совета Курганского государственного университета.

Информатика. Базовый курс : учеб. пособие / отв. ред. Ю. В. Адаменко. – Курган : Изд-во Курганского гос. ун-та, 2017. – 166 с.

В учебном пособии рассмотрены основные положения современной информатики как комплексной научно-технической дисциплины: структура и общие свойства информации и информационных процессов, принципы построения вычислительных устройств, систем обработки, хранения и передачи информации, проектирование информационных систем, защита информации в автоматизированных системах. Изложены определения ключевых понятий и конкретные вопросы по темам курса.

Учебное пособие рассчитано на студентов младших курсов, бакалавров, школьников и всех, самостоятельно изучающих информатику.

Составители:

Томилова Елена Николаевна – старший преподаватель кафедры ИТ и МПИ КГУ (раздел 1);

Говорков Андрей Викторович – канд. пед. наук, доцент, зав. кафедры ИТ и МПИ КГУ, *Козлов Сергей Олегович* – ассистент кафедры ИТ и МПИ КГУ (раздел 2);

Тетюшева Светлана Геннадьевна – старший преподаватель кафедры ИТ и МПИ КГУ (раздел 3);

Медведев Аркадий Андреевич – канд. пед. наук, доцент кафедры ИТ и МПИ КГУ, *Казанцев Дмитрий Алексеевич* – ассистент кафедры ИТ и МПИ КГУ (раздел 4);

Никифорова Татьяна Анатольевна – канд. пед. наук, доцент кафедры ИТ и МПИ КГУ (раздел 5);

Сидорова Оксана Аркадьевна – старший преподаватель кафедры ИТ и МПИ КГУ (раздел 6);

Томилова Елена Николаевна, Адаменко Юлия Владимировна (раздел 7);

Сидорова Оксана Аркадьевна, Тетюшева Светлана Геннадьевна (раздел 8).

ISBN 978-5-4217-0425-6

УДК 681.3 (075.8)

ББК 32.97 я7

©Курганский
государственный
университет, 2017

1 ИНФОРМАТИКА И ИНФОРМАЦИЯ

1.1. ИНФОРМАТИКА

1.1.1 Предмет и задачи информатики

Термин *информатика* (франц. Informatique) происходит от французских слов *information* (информация) и *automatique* (автоматика) и означает «информационная автоматика». Широко распространен также англоязычный вариант этого термина – *Computer science* (компьютерные науки).

Информатика – это основанная на использовании компьютерной техники дисциплина, изучающая структуру и общие свойства информации, а также закономерности и методы её создания, хранения, поиска, преобразования, передачи и применения в различных сферах человеческой деятельности.

В 1978 г. международный научный конгресс официально закрепил за понятием «информатика» области, связанные с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая компьютеры и их программное обеспечение, а также организационные, коммерческие, административные и социально-политические аспекты компьютеризации – массового внедрения компьютерной техники во все области жизни людей. Таким образом, информатика базируется на компьютерной технике и немыслима без нее.

Информатика изучает то общее, что свойственно всем многочисленным разновидностям конкретных информационных процессов. Эти информационные процессы и технологии и есть объект информатики. Предмет информатики определяется многообразием её приложений.

Информатика – научная дисциплина с широким диапазоном применения. Её основные направления:

- разработка вычислительных систем и программного обеспечения;
- теория информации, изучающая процессы, связанные с передачей, приемом, преобразованием и хранением информации;
- методы искусственного интеллекта, позволяющие создавать программы для решения задач, требующих определенных интеллектуальных усилий при выполнении их человеком (логический вывод, обучение, понимание речи, визуальное восприятие, игры и др.);
- системный анализ, заключающийся в анализе назначения проектируемой системы и в установлении требований, которым она должна отвечать;
- методы машинной графики, анимации, средства мультимедиа;
- средства телекоммуникаций, в том числе глобальные компьютерные сети, объединяющие все человечество в единое информационное сообщество;
- разнообразные приложения, охватывающие производство, науку, образование, медицину, торговлю, сельское хозяйство и все другие виды хозяйственной и общественной деятельности.

1.1.2 Истоки и предпосылки информатики

Появление и начальное становление информатики как науки относится ко второй половине XX века. После второй мировой войны получила бурное развитие *кибернетика* как общая наука об управлении и связи в системах различной природы – искусственных, биологических, социальных.

Рождение кибернетики принято связывать с опубликованием в 1948 г. американским математиком Норбертом Винером книги «Кибернетика, или контроль и коммуникации у животных и машин». В этой работе были показаны пути создания общей теории управления и заложены основы методов рассмотрения проблем управления и связи для различных систем с единой точки зрения. Развиваясь одновременно с прогрессом электронных вычислительных машин, кибернетика со временем превращалась в более общую науку о преобразовании информации. Под информацией в кибернетике понимается любая совокупность сигналов, воздействий или сведений, которые некоторая система воспринимает от окружающей среды (входная информация X), выдает в окружающую среду (выходная информация Y), а также хранит в себе (внутрисистемная информация Z). Визуально данное понятие представлено на рисунке 1.1.

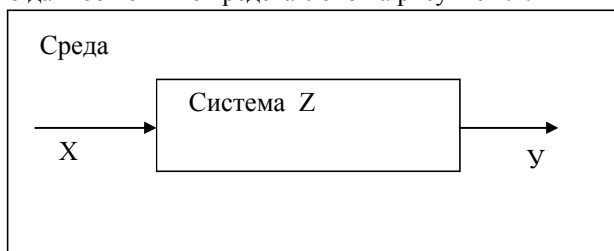


Рисунок 1.1 – Модель понятия «информация»

Название «кибернетика» происходит от греческого «кибернетес», что первоначально означало «рулевой», «кормчий», но впоследствии стало обозначать и «правитель над людьми». Так, древнегреческий философ Платон в своих сочинениях в одних случаях называет кибернетикой искусство управления кораблем или колесницей, а в других – искусство править людьми. Примечательно, что римлянами слово «кюбернетес» было преобразовано в «губернатор».

Развитие кибернетики в нашей стране переживало драматические периоды. Развитие отечественной кибернетики в течение длительного периода сопровождалось серьезными неудачами в реализации крупных государственных проектов. К таким, в первую очередь, относятся программы создания автоматизированных систем управления (АСУ), систем автоматического проектирования (САПР), гибких автоматизированных производств и промышленных роботов. Работы в указанных направлениях привели к накоплению значительного опыта создания информационных систем управления технико-экономическими объектами, но первоначально поставленные глобальные цели не были достигнуты.

Одновременное развитие кибернетики и электронно-вычислительных машин предопределило, что эта наука со временем реорганизовалась к более общей, изучающей преобразование информации в принципе. Таким образом, эволюционируя и всячески развиваясь, из кибернетики выделилась информатика, основным предметом рассмотрения которой стало непосредственно понятие информации, а также методы её обработки.

1.2 ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

1.2.1 Различные подходы к понятию «информация»

Информация – слово, которое мы читаем, слышим и употребляем в разговоре даже на бытовые темы. В любой текст это слово ввести легко: стоит только заменить обычные слова «сообщение», «данные», «новости», «сведения» на «информация». Несмотря на то, что это понятие очень широко используется и в науке, и в повседневной жизни, его строгого научного определения до настоящего времени нет. По сей день разные научные дисциплины вводят это понятие по-разному.

«Информация – сведения об окружающем мире и протекающих в нем процессах, воспринимаемые человеком или специальным устройством» [8].

«Информация – совокупность сведений, уменьшающих неопределенность в выборе различных возможностей» [5].

«Информация – сведения (сообщения, данные) независимо от формы их представления» [1].

Можно выделить три возможных подхода к понятию «информация»: антропоцентрический, техноцентрический и недетерминированный.

Суть антропоцентрического подхода состоит в том, что информацию отождествляют со сведениями или фактами, которые теоретически могут быть получены и усвоены, т.е. преобразованы в знания. Между информацией и знаниями имеется разрыв. Человек должен творчески перерабатывать информацию, чтобы получить новые знания.

Суть техноцентрического подхода состоит в том, что информацию отождествляют с данными. Данные – сведения, представленные в определенной знаковой системе и на определенном носителе для обеспечения возможностей их хранения, передачи, приема и обработки. Данные безотносительны к содержанию информации.

Недетерминированный подход состоит в отказе от определения информации на том основании, что оно является фундаментальным, как, например, материя или энергия.

Отсутствие единого понятия позволяет использовать различные критерии для классификации информации.

Первая классификация информации делит ее *по определенному типу знания*, которое приобретается. Декларативные знания – общеизвестные факты, которые доступны каждому. К примеру, такими знаниями считается следую-

шая информация: «Москва – столица нашей Родины», «Луна – это спутник нашей планеты Земля» и так далее. Таким образом, декларативные знания могут содержать в себе следующие типы информации: об объектах, событиях, свойствах объектов, о математических и других зависимостях. Процедурные знания – описывают последовательность действия для выполнения определенного задания, то есть дают инструкции. Такая информация чаще всего представляет возможность найти ответы на различные вопросы, например, «Как сварить кашу?» или «Как шить юбку?».

Вторая классификация делит всю информацию *по способам ее восприятия*, ведь большинство новых фактов и явлений мы познаем именно через органы чувств. Современные ученые высчитали, что более 80% информации человек получает при помощи зрения, 10% – слуха, 4% – обоняния, 3% – вкусовых ощущений, 2% – осязания. Но при этом не стоит забывать, что каждый человек представляет собой индивидуальность, а это значит, что у художников будет намного лучше развит именно зрительный канал, а у музыкантов эффективнее будет работать на восприятие слуховой канал.

По способу передачи и восприятия различают следующие виды информации:

- визуальную, передаваемую видимыми образами и символами;
- аудиальную, передаваемую звуками;
- тактильную, передаваемую ощущениями;
- органолептическую, передаваемую запахами и вкусами;
- машинную, выдаваемую и воспринимаемую средствами вычислительной техники.

Третья классификация в качестве критерия использует *способ представления данных*. Исходя из этого, выделяют следующие группы: текстовую, числовую, графическую, видео и звуковую информацию.

Четвертая классификация информации делит объект исследования в зависимости от его *общественного значения*. Таким образом, информация может быть общественной, специальной и личной. В свою очередь, общественная делится на быденную, общественно-политическую, научно-популярную, новостную, эстетическую. Специальная информация тоже имеет свои подвиды: производственная, управленческая, техническая и научная.

Классификация *по форме представления* подразделяет информацию на 2 вида:

- дискретная форма представления информации – это последовательность символов, характеризующая прерывистую, изменяющуюся величину (количество дорожно-транспортных происшествий, количество тяжких преступлений и т.п.);
- аналоговая или непрерывная форма представления информации – это величина, характеризующая процесс, не имеющий перерывов или промежутков (температура тела человека, скорость автомобиля на определенном участке пути и т.п.).

По области возникновения выделяют информацию:

- элементарную (механическую), которая отражает процессы, явления неодушевленной природы;
- биологическую, которая отражает процессы животного и растительного мира;
- социальную, которая отражает процессы человеческого общества.

Современная наука насчитывает более восьмидесяти различных определений понятия «информация». Классификация информации содержит в себе множество видов и типов знания, но именно с ее помощью очень легко разобраться в любой науке или сфере деятельности.

1.2.2 Свойства информации

Информация имеет свои свойства. Во-первых, она может быть *полезной* и *бесполезной*. Но так как границы между этими понятиями нет, то следует говорить о степени полезности применительно к нуждам конкретной информационной системы – работника, учреждения. Эта степень хорошо проявляется на примерах новостных сайтов. Если мы просматриваем один, самый интересный для нас новостной сайт, то не все публикуемые там материалы одинаково интересны. Можно зайти на другие новостные сайты, но там количество неинтересных материалов будет еще больше. И сколько бы сайтов мы не посетили, всегда за пределами просмотра останется много сайтов, общий объем интересного материала в которых будет достаточно большим.

Чтобы решить поставленную перед нею задачу, система должна иметь полную информацию. Под *полнотой* информации понимается её достаточность для принятия решения. Но информация никогда не бывает полной. Поэтому опять-таки следует иметь в виду степень полноты. С понятием полноты сталкиваются все, кому приходится выполнять служебные задания. Если исходные данные неполны, принять верное решение непросто. Однако, бывают случаи, когда данные полны, а верное решение мы все равно получить не можем. Это свидетельствует о том, что мы не располагаем какими-то методами. И в том, и в другом случае можно говорить о том, что недостаточно необходимой информации.

Еще одно свойство – *достоверность* информации. Под достоверностью информации понимается её соответствие объективной реальности. Недостоверность информации может быть связана с тем, что данные изначально были подготовлены как ложные (дезинформация), в результате модификации данных или в результате того, что данные трудно выделить на фоне регистрации посторонних сигналов. Известны случаи, когда недостоверные данные могут давать достоверную информацию, например, когда заранее известна степень их недостоверности.

С достоверностью информации иногда ошибочно путают другое свойство – *адекватность* информации. Под адекватностью понимают степень соответствия информации, полученной потребителем, тому, что автор вложил в

её содержание. Чтобы понять разницу этих понятий, рассмотрим рисунок 1.2.

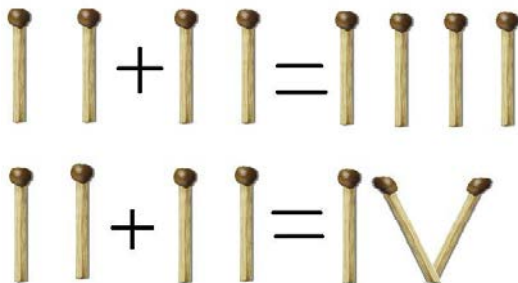


Рисунок 1.2 – Иллюстрация свойства достоверности и адекватности информации

На первом рисунке четыре спички слева и четыре таких же спички справа уравнены. Это соответствует объективной реальности и тем самым отражает свойство достоверности. На втором же рисунке слева и справа представлено разное количество спичек и объективно между ними нельзя ставить знак равенства. Однако, зная правило записи римских цифр, мы получаем адекватную информацию о том, что $2 + 2 = 4$.

Чтобы собрать полную и достоверную информацию, требуется время, а информация стареет. Поэтому рассмотрим еще одно свойство – *новизну*. Чтобы стать знанием, сведения обязательно должны содержать определенную степень новизны. Но всякое новое основывается на чем-то старом. Рассмотрим следующие примеры:

«Москва – столица нашей Родины» – является ли это сообщение информацией? Что нового мы узнали, прочитав его еще раз? Какие знания почерпнули? Никаких, так как данный факт известный всем с самого детства.

Другой случай – дадим первокласснику учебник по высшей математике. То, что в учебнике собрано много информации – факт, но сколько информации получит ученик первого класса? Нисколько, так как абсолютная новизна информации не дает оснований закрепить эти сведения в виде знаний. Вывод - новое без старого не воспринимается, так же как старое без нового тоже не воспринимается.

Чтобы информация дошла до того, для кого она предназначена, её повторяют – этим создается *избыточность*. Избыточность – гарантированная полнота восприятия. Наш естественный язык обладает избыточностью, благодаря которой мы без напряжения воспринимаем речь, несмотря на пропуски фраз, слов во фразах, и букв в словах.

Обычный текст, напечатанный на русском языке имеет избыточность порядка 20 - 25%. *Попробуйте отбросить каждую пятую букву, и вы увидите, что получите информацию из почти того же текста, хотя читать его будет очень утомительно.* (Попробуйте отбросить каждую пятую букву, и вы увидите, что получить ин-

формацию из печатного текста все же можно, хотя читать его будет очень утомительно). Нам нередко приходится иметь дело с небрежным рукописным почерком. Избыточность информации, заключенной в тексте, оказывает добрую службу, позволяя догадываться о значении неразборчивых символов.

Визуальная информация имеет избыточность более 90%. Еще большую избыточность имеет видеoinформация – 98 - 99%. Эта избыточность позволяет нам рассеивать внимание, что часто воспринимается как отдых при просмотре кинофильма. Особое значение избыточность информации имеет в информационных технологиях, ориентированных на автоматическую обработку данных.

Доступность информации – это мера возможности получить ту или иную информацию. Доступность информации обеспечивается за счёт её преобразования в понятную форму. При этом одну и ту же информацию можно представить в разной форме в зависимости от её получателя. Информация становится понятной, если она выражена в той форме и тем языком, который воспринимает тот, кому она предназначена.

Актуальность – это степень соответствия информации текущему моменту времени. Это свойство во многом зависит от интервала времени, прошедшего с момента появления данной информации, а также от того, насколько быстро изменяется ситуация. Фактор времени в определении актуальности информации в ряде случаев является очень важным, поскольку некоторые виды информации и информационных услуг имеют смысл только в определенный промежуток времени. Например, получение заранее заказанного билета на самолет после его вылета теряет всякий смысл. Точно так же получение прогноза погоды на вчерашний день не имеет никакого смысла, поскольку это событие уже наступило. В этом контексте весьма уместной является поговорка: «Дорога ложка к обеду». Таким образом, своевременность информации предполагает её поступление не позже заранее назначенного момента времени, согласованного со временем решения поставленной задачи. Только актуальная, вовремя полученная информация может принести пользу людям.

Репрезентативность информации связана с правильностью её отбора и формирования для адекватного отражения свойств объекта. Непременным условием определения свойства репрезентативности информации является поступление похожей информации из разных источников. Понятно, что полного совпадения по всем источникам информации никогда не будет. Однако если все сделано правильно, то полученная информация будет отражать самые важные характеристики объекта. Важнейшее значение здесь имеют выбранные методики и методы отбора информации и её оценки.

Объективность – это характеристика информации, выражающая степень её соответствия реальной действительности.

Объективность – свойство информации, которое, к сожалению, крайне редко бывает абсолютным. Казалось бы, объективная и материальная природа данных, в которых информация хранится, должна давать все основания, чтобы

информация тоже была объективной. Однако так происходит далеко не всегда, ведь данные – это только один компонент информации. Второй компонент – информационные методы – связан с источником или потребителем информации и имеет субъективную природу. В зависимости от того, какой компонент превалирует в информационном процессе, результирующая информация может быть объективной более или менее.

В науке принято считать объективной информацию воспроизводимую. Например, законы физики воспроизводимы, а законы астрологии – нет. Соответственно, физика считается объективной наукой, а астрология – нет.

Основной способ повышения объективности информации заключается в увеличении ее полноты. Оценка выступления спортсмена в фигурном катании всегда субъективна, но чем больше разных арбитров, не имеющих внутренних информационных связей друг с другом, ее поставили, тем выше ее объективность.

Экзаменационная оценка тоже всегда субъективна. Поэтому в особо важных случаях экзамен сдают не одному экзаменатору, а экзаменационной комиссии. Коллегиальность принятия окончательного решения – это мера, направленная на повышение его объективности.

Не следует путать достоверность с объективностью. Необъективная информация действительно не может быть достоверной, но обратное возможно: недостоверная информация вполне может быть объективной. На этом основаны все естественные и технические науки. Мы сталкиваемся с этим явлением каждый день, когда что-либо измеряем.

Невозможно достоверно знать длину береговой линии островов и материков. Невозможно достоверно определить площадь поверхности листьев на дереве. В физике мы не можем достоверно измерить ни массу, ни длину, ни время, а потому не можем абсолютно достоверно знать значения ни одной физической константы. Но мы всегда можем абсолютно достоверно указать предельную погрешность принятого значения. Это делает математику, физику, химию, физическую географию и астрономию объективными науками.

Рассмотрим примеры, иллюстрирующие вышеперечисленные свойства.

Утром, когда вы собираетесь на занятия, вы обязательно смотрите на часы: вам нужна только достоверная информация. Скорее всего, вы выглянете в окно или посмотрите на термометр, чтобы решить, что следует надеть. Здесь важна актуальность информации. Затем вы идете на занятия и находите кабинет, согласно расписанию. Вам нужна полная и достоверная информация, иначе невозможно найти нужный кабинет.

Вам необходимо воспользоваться географической картой для определения маршрута поездки, знакомства с новой страной, изучения исторических событий. Карта всегда служила человеку источником информации о земной поверхности. Она также является важным инструментом для исследования в различных областях. Поэтому здесь жизненно важна адекватность информации, содержащейся в картах, реальной местности.

В настоящее время огромную роль играют геоинформационные системы. В них обрабатывается и анализируется информация, поступающая со спутников. Такие системы позволяют решать огромное множество задач от прогнозирования объема продаж и потенциала рынка до построения модели рельефов поверхности Земли. Все карты «описаны» специальным языком, который понятен лишь специалисту. Это означает, что данная информация доступна не всем. Каждый символ для специалиста несет большой объем достоверной, объективной и понятной информации, которая недоступна тем, кто не знает используемого языка.

В современных «космических технологиях» решающую роль играет информация, полученная с помощью различных приборов. Например, расположение станции относительно Солнца важно для работы солнечных батарей. Малейшая неточность – и космический корабль лишится энергии. Такая информация должна быть актуальной, достоверной и полной.

Кроме этого информация обладает еще следующими свойствами:

1) *атрибутивные свойства* (атрибут – неотъемлемая часть чего-либо). Важнейшими среди них являются: дискретность (информация состоит из отдельных частей, знаков) и непрерывность (возможность накапливать информацию);

2) *динамические свойства* связаны с изменением информации во времени;

3) *практические свойства* – информационный объем и плотность.

1.2.3 Измерение информации

Существует несколько подходов к измерению информации.

Первый подход – *вероятностный*.

Научный подход к оценке сообщений был предложен еще в 1928 году Р. Хартли. Расчетная формула Хартли для равновероятностных событий имеет вид:

$$I = \log_2 N \text{ или } 2^I = N, \quad (1)$$

где N – количество *равновероятных* событий (число возможных выборов);

I – количество информации.

Единицей измерения количества информации является бит.

Пример 1.1. Сколько бит информации несет сообщение о том, что поезд прибывает на один из 8 путей?

Формула Хартли: $I = \log_2 N$,

где N – число равновероятных исходов события, о котором речь идет в сообщении, ($N = 8$);

I – количество информации в сообщении.

$I = \log_2 8 = 3$ (бит).

Ответ: 3 бита.

Вероятность события вычисляется по формуле

$$p = \frac{K}{N}, \quad (2)$$

где K – величина, показывающая, сколько раз произошло интересующее нас событие;

N – общее число возможных исходов, событий.

С использованием вероятности формула (1) записывается следующим образом:

$$I = \log_2 \frac{1}{p}. \quad (3)$$

Если вероятность становится меньше, то количество информации увеличивается.

Пример 1.2. В классе 30 человек. За контрольную работу по математике получено 6 пятерок, 15 четверок, 8 троек и 1 двойка. Сколько бит информации несет сообщение о том, что Иванов получил четверку?

Вероятность интересующего нас события $p = 15/30$.

Количество информации в сообщении $I = \log_2(30/15) = \log_2 2 = 1$.

Ответ: $I = 1$ бит.

Общий случай вычисления количества информации в сообщении об одном из N , но уже неравновероятных событий был предложен К. Шенноном в 1948 году.

$$I = -\sum_{i=1}^N p_i \log_2 p_i, \quad (4)$$

где I – количество информации;

p_i – вероятность наступления i -ого события из N -возможных.

Второй подход – *алфавитный*. При этом подходе количество информации не зависит от содержания, а зависит от мощности алфавита и количества символов в тексте.

Мощность алфавита (A) – количество символов в нем. Объем сообщения измеряется по формуле:

$$V = n \cdot i, \quad (5)$$

где n – количество символов в сообщении;

i – информационный объем одного символа, который вычисляется по формуле

$$2^i = A. \quad (6)$$

Пример 1.3. Измерить информационный объем следующей фразы:

О, сколько нам открытий чудных готовит просвещения дух

Так как фраза набрана компьютерным способом, полагаем, что использовался компьютерный алфавит в кодировке ASCII. Это означает, что информационный объем каждого символа равен 8 бит/символ, т.е. $i = 8$ бит/символ. Количество символов в этой фразе – 54, т.е. $n = 54$ символа.

Подставляя эти данные в формулу для вычисления информационного объема, имеем: $V = 54$ бит/символ \cdot 8 символов = 432 бита.

Ответ: $V = 432$ бита.

Наименьшей единицей измерения количества или объема информации является величина бит. С точки зрения вероятностного подхода 1 бит – это количество информации в сообщении, которое ровно наполовину уменьшает неопределенность о том или ином событии. В алфавитном подходе 1 бит – это минимальный объем, занимаемый 0 или 1 при двоичном кодировании информации.

1.2.4 Единицы измерения информации

Выше уже было сформулировано, что с теоретической точки зрения 1 бит – это информация, которая сокращает неопределенность наших знаний вдвое (ответ на вопрос типа «да»/«нет», наличие или отсутствие какого-либо свойства, четность числа и т.д.). С точки зрения практической реализации компьютерных устройств для обработки информации 1 бит – это отдельный двоичный разряд любого из таких устройств. Иначе говоря, в вычислительной технике бит служит конструктивной базой для построения всех цифровых двоичных устройств: регистров, сумматоров и т.п. Отсюда очевидно, что в теории информации количество бит может быть любым, в том числе дробным, в то время как в реальных устройствах оно обязательно целое.

Бит, будучи минимально возможной порцией информации в компьютере, довольно маленькая единица измерения. Поэтому на практике чаще всего используется другая единица, которая называется 1 байт = 8 бит. С точки зрения устройства компьютера байт замечателен тем, что является минимальной адресуемой информацией в компьютере, иначе говоря, считать из памяти часть байта невозможно. В современных компьютерах все устройства памяти имеют байтовую структуру, а внешние устройства также обмениваются информацией байтами или кратными ему порциями. Как следствие все типы данных (числа, символы и др.) представляются в компьютере величинами, кратными байту.

С целью получения шкалы для измерения объемов информации в широких пределах от байта с помощью стандартных приставок образуется целая система более крупных производных единиц:

1 килобайт = 1024 байт

1 мегабайт = 1024 килобайт

1 гигабайт = 1024 мегабайт

1 терабайт = 1024 гигабайт

1 петабайт = 1024 терабайт и т.д.

Число $2^{10} = 1024$ достаточно близко к тысяче, используемой в качестве ос-

нования десятичных приставок СИ. Среди степеней двойки ни одна больше не близка настолько к степени десяти; к тому же показатель двоичной степени «10» сам по себе оказался удобен для грубого пересчёта двоичных степеней на привычные людям десятичные числа. Для обозначения $2^{10}=1024$ байт придумали единицу «К» (*ка*, очевидно, искажённое «кило»). Из-за близости множителей 1024 и 1000 в разговорной речи «К» всё равно называли «кило», и вскоре такая интерпретация приставки *кило* стала стандартом де-факто, как и экстраполяция на другие приставки: 1 «килобайт» = 1024 байт, 1 «мегабайт» = 1024 килобайт = 1048576 байт, и т. д.

Таким образом термины, предназначенные для десятичных приставок СИ, стали применяться к близким двоичным числам. Причём эти приставки часто используют по своему усмотрению, то есть одни понимают их как двоичные приставки, а другие как десятичные. Например, размер оперативной памяти компьютера обычно приводится в двоичных единицах (1 килобайт = 1024 байт), а размер дисков их производители указывают в десятичных (1 килобайт = 1000 байт).

IEEE 1541-2002 – стандарт Института инженеров электротехники и электроники (IEEE), содержащий рекомендации по применению двоичных приставок единиц измерения в области цифровой и вычислительной техники, предлагает близкие по написанию термины для «настоящих» двоичных мегабайтов и гигабайтов. В соответствии с этим стандартом, вместо «мегабайта» следует говорить «мебибайт», вместо «гигабайта» – «гибибайт», а вместо терабайта – тебибайт. В сокращённое наименование этих единиц предлагается вставлять букву «и»: KiB, MiB, GiB.

Однако в нашей речи еще надолго останется старое обозначение, имеющее в разных ситуациях разный смысл.

1.2.5 Информационные процессы

Информационным процессом можно назвать такой процесс, в результате которого осуществляется поиск, сбор, хранение, обработка, кодирование и передача информации. Информационный процесс может состояться только при наличии информационной системы, обеспечивающей его составляющие. Что же подразумевается под информационной системой?

Понятие «информация» обычно предполагает наличие двух объектов - источника информации и потребителя информации. Информация от источника к приемнику передается в материально-энергетической форме. Информация может поступать непрерывно, а может и дискретно, т.е. в виде последовательности отдельных сигналов, отделенных друг от друга временными или пространственными промежутками.

Информация, переносимая сигналами, имеет смысл, отличный от смысла самого факта поступления сигнала. Так, удар барабана, звуковой сигнал, мог информировать о наступлении противника. Другими словами, информация бывает о чем-то. Сигнал об этом, принимаемый потребителем, может и

не иметь прямой физической связи с событием или явлением, о котором он сигнализирует. Следовательно, поступивший сигнал должен быть воспринят адресатом и обработан, может быть даже преобразован и, возможно, с целью использования. Таким образом, можно говорить, что составляющие информационного процесса обеспечиваются наличием информационной системы. Она состоит из источника информации, канала связи, по которому информация в форме материально-энергетического сигнала (например, электрического, светового, звукового и т.д.) может поступить к потребителю, а также некоторого соглашения (кода), которое позволит потребителю установить смысл воспринятого сигнала, и собственно адресата, потребителя информации.

Установлена общность информационных процессов в живой природе, обществе и технике. Подтвердим сказанное примерами.

Известны сезонные изменения, происходящие в растительном мире – например, весной вырастают листья, которые осенью желтеют и опадают. Это результат происходящего информационного процесса. В животном мире похожая картина, но этот мир находится на более высокой ступени развития, так как у представителей животного мира есть головной мозг (или хотя бы нервная система) – орган, предназначенный для управления всеми стадиями информационных процессов, протекающих в живом организме. Так, у животных, живущих сообществами, даже выработался особый язык для передачи сообщений друг другу.

Вместе с тем можно утверждать, что всей неживой природе информационный процесс не свойственен – ни камню, ни умершему живому организму. При воздействии окружающей среды в них могут происходить различные физико-химические изменения. Но эти изменения происходят лишь в результате физического воздействия и/или химической реакции, а не какого-либо информационного процесса. Информационного процесса здесь нет и не может быть, так как отсутствует восприятие параметров внешней и внутренней среды как сигналов, подлежащих обработке, тем самым переводящее их в ранг информации, которую можно использовать, накопить, передать. Здесь нет информационной системы, обеспечивающей все составляющие информационного процесса.

В неживой природе, рассматривая информационные процессы, можно говорить лишь о технике. Техника повторяет (моделирует) некоторые действия человека и способна в этих случаях заменить его, а также используется человеком для увеличения своих природных возможностей.

Что касается человеческого общества, то оно немислимо без информационных процессов. Люди издавна объединились, что обусловило не только различные способы передачи информации от человека человеку, но и разработку средств передачи. Форма передачи информации всегда определялась возможной формой её восприятия. Но в какой бы форме и какими бы средствами не передавалась информация, она всегда передается с помощью какого-либо языка.

Применяя язык как способ представления информации, а также различ-

умножения целую часть произведения. Число в новой системе счисления записывается как последовательность полученных целых частей произведения.

Умножение производится до достижения заданной погрешности, либо до тех пор, пока дробная часть произведения не станет равной нулю или ситуация не перейдет в стадию закливания. Рассмотрим перевод на примере перевода дробных чисел 0,25 и 0,15 из десятичной в двоичную систему.

0,	25
	2
0	50
	2
1	00

0,	15
	30
0	60
1	20
0	40
0	80
1	60
1	20

$\left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \text{период}$

Ответ: $0,25_{10} = 0,01_2$ $0,15_{10} = 0,00(1001)_2$

При переводе числа из системы с основанием q в десятичную нужно это число представить в виде суммы слагаемых, каждое из которых является произведением цифры числа на основание системы q в степени, соответствующей разряду соответствующей цифры в числе.

Пример 1.4. Разряды

$$\begin{aligned} \text{Число } 1\ 0\ 1\ 1, 1_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} = \\ &= 8 + 0 + 2 + 1 + 0,5 = 11,5_{10} \end{aligned}$$

Пример 1.5. Разряды

$$\begin{aligned} \text{Число } 2\ 7\ 6, 5_8 &= 2 \times 8^2 + 7 \times 8^1 + 6 \times 8^0 + 5 \times 8^{-1} = 128 + 56 + 6 + \\ &+ 0,625 = 190,625_{10} \end{aligned}$$

Пример 1.6. Разряды

$$\begin{aligned} \text{Число } 1\ 13\ 11, 3_{16} &= 1 \times 16^2 + 11 \times 16^1 + 3 \times 16^0 = 256 + 176 + 3 = \\ &= 435_{10} \end{aligned}$$

Перевод чисел из двоичной системы счисления в восьмеричную.

Каждый разряд восьмеричного числа содержит 3 бита информации. Таким образом, для перевода целого двоичного числа в восьмеричное его нужно разбить на группы по три цифры, справа налево, а затем преобразовать каждую группу в восьмеричную цифру. Если в последней, левой, группе окажется меньше трех цифр, то необходимо ее дополнить слева нулями.

Переведем таким способом двоичное число 101001_2 в восьмеричное:

$$101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5 \quad 001_2 = 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1$$

Следовательно, $101001_2 = 51_8$.

Для перевода дробного двоичного числа (правильной дроби) в восьмеричное необходимо разбить его на триады слева направо и, если в последней, правой, группе окажется меньше трех цифр, дополнить ее справа нулями. Да-

лее необходимо триады заменить на восьмеричные числа.

Пример 1.7. Преобразуем дробное двоичное число $0,110101_2$ в восьмеричную систему счисления:

<i>Двоичные триады</i>	<i>110</i>	<i>101</i>
<i>Восьмеричные цифры</i>	<i>6</i>	<i>5</i>

Получаем: $0,110101_2 = 0,65_8$.

Перевод чисел из двоичной системы счисления в шестнадцатеричную.

Каждый разряд шестнадцатеричного числа содержит 4 бита информации. Таким образом, для перевода целого двоичного числа в шестнадцатеричное его нужно разбить на группы по четыре цифры (тетрады), начиная справа, и, если в последней левой группе окажется меньше четырех цифр, дополнить ее слева нулями.

Для перевода дробного двоичного числа (правильной дроби) в шестнадцатеричное необходимо разбить его на тетрады слева направо и, если в последней правой группе окажется меньше четырех цифр, то необходимо дополнить ее справа нулями.

Затем надо преобразовать каждую группу в шестнадцатеричную цифру.

Пример 1.8. Переведем целое двоичное число 101001_2 в шестнадцатеричное:

<i>Двоичные тетрады</i>	<i>0010</i>	<i>1001</i>
<i>Шестнадцатеричные цифры</i>	<i>2</i>	<i>9</i>

В результате имеем: $101001_2 = 29_{16}$.

Переведем дробное двоичное число $0,110101_2$ в шестнадцатеричную систему счисления:

<i>Двоичные тетрады</i>	<i>1101</i>	<i>0100</i>
<i>Шестнадцатеричные цифры</i>	<i>$13_{10} = D_{16}$</i>	<i>4</i>

Получаем: $0,110101_2 = 0,D4_{16}$.

Для того чтобы преобразовать любое двоичное число в восьмеричную или шестнадцатеричную системы счисления, необходимо произвести преобразования по рассмотренным выше алгоритмам отдельно для его целой и дробной частей.

Перевод чисел из восьмеричной и шестнадцатеричной систем счисления в двоичную.

Для перевода чисел из восьмеричной и шестнадцатеричной систем счисления в двоичную необходимо цифры числа преобразовать в группы двоичных цифр. Для перевода из восьмеричной системы в двоичную каждую цифру числа надо преобразовать в группу из трех двоичных цифр (триаду), а при преобразовании шестнадцатеричного числа – в группу из четырех цифр (тетраду).

Пример 1.9. Преобразуем целое восьмеричное число 47_8 в двоичную систему счисления:

<i>Восьмеричные цифры</i>	<i>4</i>	<i>7</i>
<i>Двоичные триады</i>	<i>100</i>	<i>111</i>

Получаем: $47_8 = 100111_2$.

Переведем целое шестнадцатеричное число AB_{16} в двоичную систему счисления:

<i>Шестнадцатеричные цифры</i>	<i>A</i>	<i>B</i>
<i>Двоичные тетрады</i>	1010	1011

В результате имеем: $AB_{16} = 10101011_2$.

1.3.2 Кодирование символьной информации

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

Институт стандартизации США (ANSI – American National Standard Institute) ввел в действие систему кодирования ASCII (читается *аски*). Данная система представляет собой таблицу, в которой каждому символу компьютерного алфавита ставится в соответствие определенный числовой код. Таблица состоит из двух частей – базовой и расширенной, как представлено в таблице 1.2.

Таблица 1.2 – Структура ASCII-таблицы

Порядковый номер	Числовой код	Символы	
0 ... 31	00000000 ... 00011111	В этой области размещаются управляющие коды, которым не соответствуют никакие символы языка и, соответственно, эти коды не выводятся ни на экран, ни на устройство печати, но ими можно управлять тем, как производится вывод прочих данных.	Базовая часть
32 ... 127	00100000 ... 01111111	В этой области размещаются коды символов английского алфавита, знаков препинания, цифр, арифметических действий и некоторых вспомогательных символов.	
128 ... 255	10000000 ... 11111111	В этой области размещаются коды национальных систем кодирования.	Расширенная часть

Поддержка производителей оборудования и программ вывела американский код ASCII на уровень международного стандарта. Международным стандартом является лишь первая половина таблицы, что же касается расширенной части, то отсутствие единого стандарта в этой области привело к множествен-

ности одновременно действующих кодировок. К сожалению, в настоящее время существуют пять различных кодировок кириллицы (КОИ8-Р, Windows, MS-DOS, Macintosh и ISO). Из-за этого часто возникают проблемы с переносом русского текста с одного компьютера на другой, из одной программной системы в другую.

Хронологически одним из первых стандартов кодирования русских букв на компьютерах был КОИ8 (Код обмена информацией, 8-битный). Эта кодировка применялась еще в 70-ые годы на компьютерах серии ЕС ЭВМ, а с середины 80-х стала использоваться в первых русифицированных версиях операционной системы UNIX.

От начала 90-х годов, времени господства операционной системы MS DOS, остается кодировка CP866 («CP» означает Code Page, кодовая страница).

Компьютеры фирмы Apple, работающие под управлением операционной системы Mac OS, используют свою собственную кодировку Mac-CP10007.

Кроме того, Международная организация по стандартизации (International Standards Organization, ISO) утвердила в качестве стандарта для русского языка еще одну кодировку под названием ISO 8859-5.

Наиболее распространенной в настоящее время является кодировка Microsoft Windows, обозначаемая сокращением CP1251.

Если проанализировать организационные трудности, связанные с созданием единой системы кодирования текстовых данных, то можно прийти к выводу, что они вызваны ограниченным набором кодов (256). В то же время очевидно, что если кодировать символы не восьмиразрядными двоичными числами, а числами с большим количеством разрядов, то и диапазон возможных значений кодов станет намного больше. Такая система, основанная на 16-разрядном кодировании символов, получила название универсальной – UNICODE. 16 разрядов позволяют обеспечить уникальные коды для 65536 различных символов. Этого количества достаточно для размещения в одной таблице символов большинства языков планеты.

1.3.3 Кодирование графической информации

Графическая информация, представленная в виде рисунков, фотографий, слайдов, подвижных изображений (анимация, видео), схем, чертежей, может создаваться и редактироваться с помощью компьютера, при этом она соответствующим образом кодируется. В настоящее время существует достаточно большое количество прикладных программ для обработки графической информации, но все они реализуют два вида компьютерной графики – растровую и векторную.

Если более пристально рассмотреть графическое изображение на экране монитора компьютера, то можно увидеть большое количество разноцветных точек, которые, будучи собраны вместе, и образуют данное графическое изображение. Из этого можно сделать вывод: графическое изображение в компьютере определенным образом кодируется и должно быть представлено в виде

графического файла. Файл является основной структурной единицей организации и хранения данных в компьютере и в данном случае должен содержать информацию о том, как представить этот набор точек на экране монитора.

Файлы, созданные на основе векторной графики, содержат информацию в виде математических зависимостей (математических функций) и соответствующих данных о том, как построить изображение объекта с помощью отрезков линий (векторов) при выводе его на экран монитора компьютера.

Файлы, созданные на основе растровой графики, предполагают хранение данных о каждой отдельной точке изображения. Для отображения растровой графики не требуется сложных математических расчетов, достаточно лишь получить данные о каждой точке изображения (ее координаты и цвет) и отобразить их на экране монитора компьютера.

В процессе кодирования изображения производится его пространственная дискретизация, т. е. изображение разбивается на отдельные точки и каждой точке задается код цвета (желтый, красный, синий и т. д.). Один такой фрагмент называется пикселем. Пиксель – минимальный участок изображения, к которому независимым образом можно задать цвет.

Для кодирования каждого пикселя цветного графического изображения применяется принцип декомпозиции произвольного цвета на основные его составляющие, в качестве которых используют три основных цвета: красный (английское слово *Red*, обозначают буквой *R*), зеленый (*Green*, обозначают буквой *G*), синий (*Blue*, обозначают буквой *B*). Любой цвет точки, воспринимаемый человеческим глазом, можно получить путем аддитивного (пропорционального) сложения (смешения) трех основных цветов – красного, зеленого и синего. Такая система кодирования называется цветовой системой *RGB*. Файлы графических изображений, в которых применяется цветовая система *RGB*, представляют каждую точку изображения в виде трех шестнадцатеричных числовых величин *R*, *G* и *B*, соответствующих интенсивностям красного, зеленого и синего цветов. Например, AAC1D5 означает, что интенсивность красного – $AA_{16} = 170_{10}$, интенсивность зеленого – $C1_{16} = 193_{10}$, интенсивность синего цвета – $D5_{16} = 213_{10}$.

Если для кодирования яркости каждой из основных составляющих использовать по 256 значений, то на кодирование цвета одного пикселя необходим объем в 24 бита. При этом система кодирования обеспечивает однозначное определение 16777216 цветов, что близко к чувствительности человеческого глаза.

Каждому из основных цветов можно поставить в соответствие дополнительный цвет, т.е. цвет, дополняющий основной до белого. Дополнительными цветами являются: голубой (Cyan, C), пурпурный (Magenta, M), желтый (Yellow, Y). Принцип декомпозиции произвольного цвета на составляющие компоненты можно применять и для дополнительных цветов, т.е. любой цвет можно представить в виде суммы голубой, пурпурной и желтой составляющей. Такой метод кодирования цвета принят в полиграфии, где используется еще и четвер-

тая краска – черная (Black, K). Поэтому данная система кодирования обозначается четырьмя буквами – CMYK, и для цвета одного пикселя в этой системе необходимо уже 32 двоичных разряда.

Качество растрового изображения определяется двумя основными параметрами – разрешением (количеством точек по горизонтали и вертикали) и используемой палитрой цветов (количеством задаваемых цветов для каждой точки изображения). Разрешение задается указанием числа точек по горизонтали и по вертикали, например 800 на 600 точек.

Между количеством цветов, задаваемых точке растрового изображения, и количеством информации, которое необходимо выделить для хранения цвета точки, существует зависимость, определяемая формулой Р. Хартли (1).

Количество информации, необходимое для хранения цвета точки, называют также глубиной цвета, или качеством цветопередачи.

Так, если количество цветов, задаваемых для точки изображения, $N = 256$, то количество информации, необходимое для ее хранения (глубина цвета), в соответствии с формулой (1) будет равно $I = 8$ бит.

Если в растровой графике базовым элементом изображения является точка, то в векторной графике – линия. *Линия* – элементарный объект векторной графики. Как и любой объект, линия обладает свойствами: формой (прямая, кривая), толщиной, цветом, начертанием (сплошная, пунктирная). Замкнутые линии приобретают свойство заполнения. Охватываемое ими пространство может быть заполнено другими объектами (текстуры, карты) или выбранным цветом. Все прочие объекты векторной графики состояются из линий. Например, куб можно составить из шести связанных прямоугольников, каждый из которых, в свою очередь, образован четырьмя связанными линиями. Возможно, представить куб и как двенадцать связанных линий, образующих ребра.

Компьютер хранит элементы изображения (линии, кривые, фигуры) в виде математических формул. При открытии файла программа прорисовывает элементы изображения по их математическим формулам (уравнениям).

Векторную графику часто называют объектно-ориентированной или чертежной графикой. Имеется ряд простейших объектов (примитивов): точка, прямая, кривая второго порядка, кривая третьего порядка, кривая Безье. Эти примитивы и их комбинации используются для создания более сложных изображений. Если посмотреть содержание файла векторной графики, обнаруживается сходство с программой. Он может содержать команды, похожие на слова, и данные в коде ASCII, поэтому векторный файл можно отредактировать с помощью текстового редактора.

Описание окружности (в упрощенном виде):

- объект – окружность;
- центр – 50, 70;
- радиус – 40;
- линия: цвет – черный, толщина – 0.50;
- заливка – нет.

Данный пример показывает основное достоинство векторной графики – описание объекта является простым и занимает мало памяти. Для описания этой же окружности средствами растровой графики потребовалось бы запомнить каждую отдельную точку изображения, что заняло бы гораздо больше памяти.

Преимущества векторной графики по сравнению с растровой:

- простота масштабирования изображения без ухудшения его качества;
- независимость объема памяти, требуемой для хранения изображения, от выбранной цветовой модели.

Недостаток: некоторая искусственность – любое изображение необходимо разбить на конечное множество составляющих его примитивов.

Векторные рисунки могут включать в себя и растровые изображения. Векторные и растровые изображения могут быть преобразованы друг в друга.

Фрактальная графика – одна из быстроразвивающихся и перспективных видов компьютерной графики. Математическая основа – фрактальная геометрия. Фрактал – структура, состоящая из частей, подобных целому. Одним из основных свойств является самоподобие (фрактус – состоящий из фрагментов).

Объекты называются самоподобными, когда увеличенные части объекта походят на сам объект. Небольшая часть фрактала содержит информацию о всем фрактале. Фрактальная графика основана на математических вычислениях. Базовым элементом фрактальной графики является сама математическая формула, то есть никаких объектов в памяти компьютера не хранится и изображение строится исключительно по уравнениям. Примеры фракталов приведены на рисунке 1.3.

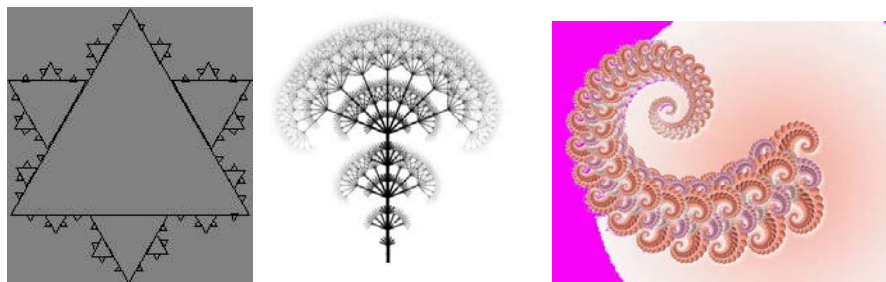


Рисунок 1.3 – Примеры фракталов

1.3.4 Кодирование звуковой информации

Звук представляет собой распространяющуюся в воздухе, воде или иной среде волну, т. е. является аналоговым сигналом. Компьютер же обрабатывает дискретные сигналы, поэтому для кодирования и обработки звуковой информации непрерывный звуковой сигнал должен быть преобразован в цифровую дискретную форму. Этот процесс называется временной дискретизацией. Не-

прерывная звуковая волна разбивается на отдельные маленькие временные участки, для каждого такого участка устанавливается определенная величина интенсивности звука. Таким образом, непрерывная зависимость громкости звука от времени $A(t)$ заменяется на дискретную последовательность уровней громкости. На рисунке 1.4 это выглядит как замена гладкой кривой на последовательность «ступенек».

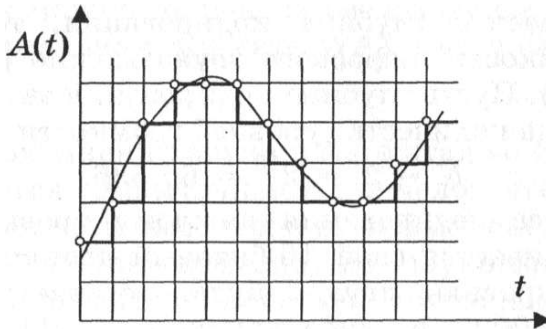


Рисунок 1.4 – Временная дискретизация звука

Качество полученного цифрового звука зависит от количества измерений уровня громкости звука в единицу времени, т. е. *частоты дискретизации*. Чем большее количество измерений производится за 1 секунду (чем больше частота дискретизации), тем точнее «лесенка» цифрового звукового сигнала повторяет кривую диалогового сигнала. Частота дискретизации измеряется в Герцах.

Каждой «ступеньке» присваивается определенное значение уровня громкости звука. Уровни громкости звука можно рассматривать как набор возможных состояний N , для кодирования которых необходимо определенное количество информации I , которое называется глубиной кодирования звука.

Глубина кодирования звука – это количество информации, которое необходимо для кодирования дискретных уровней громкости цифрового звука.

Если известна глубина кодирования, то количество уровней громкости цифрового звука можно рассчитать по формуле $N = 2^I$. Пусть глубина кодирования звука составляет 16 битов, тогда количество уровней громкости звука равно:

$$N = 2^I = 2^{16} = 65\,536.$$

Чем больше частота и глубина дискретизации звука, тем более качественным будет звучание оцифрованного звука. Самое низкое качество оцифрованного звука, соответствующее качеству телефонной связи, получается при частоте дискретизации 8000 раз в секунду, глубине дискретизации 8 битов и записи одной звуковой дорожки (режим «моно»). Самое высокое качество оцифрованного звука, соответствующее качеству аудио-CD, достигается при частоте дискретизации 48000 раз в секунду, глубине дискретизации 16 битов и

записи двух звуковых дорожек (режим «стерео»).

Некоторые из используемых частот дискретизации звука:

- 8 000 Гц – телефон, достаточно для речи;
- 11 025 Гц – достаточно для передачи речи;
- 22 050 Гц – достаточно для передачи качества радио;
- 44 100 Гц – используется в Audio CD;
- 48 000 Гц – DVD.

1.4 Контрольные вопросы

1 В чем причина проблемы определения понятия «информация»? Какие возможны подходы к определению информации?

2 Как вы можете объяснить бытовой термин «переизбыток информации»?

3 Запишите формулу Хартли, объясните значение используемых в ней величин.

4 Что такое информационные процессы? Приведите примеры информационных процессов из литературных источников, определите в них источник, приемник и канал связи.

5 Приведите примеры сообщений, несущих 1 бит информации.

6 Как вы понимаете динамический характер информации? Что происходит с ней по окончании информационного процесса?

7 Приведите примеры систем счисления, назовите их основание и перечислите цифровой набор этих систем.

8 К каким группам символов относятся символы с кодами 12, 105, 216 согласно таблице ASCII?

9 Что такое растровое и векторное изображение? Назовите достоинства и недостатки каждого из них.

10 Объясните суть принципов формирования цвета RGB, CMYK.

11 Объясните, что такое пространственная и временная дискретизация.

12 Как определить количество уровней громкости звукового сигнала, если известна глубина кодирования?

13 Объем сообщения составляет 65536 бит. Переведите это значение в байты, килобайты, мегабайты.

Список источников

1 Об информации, информационных технологиях и о защите информации [Электронный ресурс] : федер. закон от 27 июля 2006 г., № 149-ФЗ // КонсультантПлюс : справ. правовая система.

2 Возможное и невозможное в кибернетике : сб. ст. / под ред. А. Берга, Э. Кольмана. – Москва : Наука, 1964. – 223 с.

3 Информатика для юристов и экономистов : учебник для вузов / С. В. Симонович [и др.]. – 2-е изд. – Санкт-Петербург : Питер, 2013. – 544 с.

4 Лапчик М. П. Методика преподавания информатики : учеб. пособие для

студ. пед. вузов / М. П. Лапчик, И. Г. Семакин, Е. К. Хеннер ; под общ. ред. М. П. Лапчика. – 4-е изд., стер. – Москва : Изд. центр «Академия», 2007. – 624 с.

5 Микиша А. М., Орлов В. Б. Толковый математический словарь. Основные термины : около 2500 терминов. – Москва : Рус. яз., 1989. – 244 с.

6 Савенкова Т. И. Информатика на пять. URL : <http://www.5byte.ru>.

7 Сизоненко М. Классификация информации. URL : <http://fb.ru/article/45331/klassifikatsiya-informatsii>.

8 Толковый словарь русского языка для школьников. – Москва : ЗАО «Славянский дом книги», 1998.

9 Шауцукова Л. З. Информатика : учеб пособие для 10-11 кл. общеобразоват. учреждений. – Москва : Просвещение, 2004. – 416 с.

2 ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

2.1 ИСТОРИЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

2.1.1 Нулевое поколение – механические компьютеры (1642-945)

Французский ученый Блез Паскаль (1623 – 1662), в честь которого был назван один из распространенных языков программирования, был первым человеком, создавшим механическую *счетную машину*. Машина была им сконструирована в 1642 году, когда ему было всего 19 лет. Машина предназначалась для его отца, сборщика налогов. На рисунке 2.1 вы можете увидеть, как выглядит эта машина. Это была механическая конструкция с шестеренками и ручным приводом.



Рисунок 2.1 – Счетная машина

Возможности машины были небольшими: она могла только складывать и вычитать. Спустя тридцать лет великий немецкий математик барон Готфрид Вильгельм фон Лейбниц (1646 – 1716) построил свою механическую машину, которая могла то же, что и машина Паскаля, а также могла умножать и делить. Таким образом, Лейбниц уже три века назад создал прибор напоминающий карманный калькулятор с четырьмя функциями.

Спустя полтора века профессор математики Кембриджского университета, Чарльз Бэббидж (1792 – 1871) разработал и сконструировал две машины: *аналитическую* и *разностную* (рисунок 2.2). Преимущество аналитической машины заключалось в том, что она могла исполнять разные задания. В отличие от раз-

ностной аналитическая машина могла выполнять несколько алгоритмов. Она считывала команды с перфокарт и выполняла их. Некоторые команды приказывали машине взять два числа из памяти, перенести их в вычислительное устройство, произвести над ними операцию (например, сложить) и отправить результат обратно в запоминающее устройство. Были также и другие команды. Так, например, была команда, которая брала число, потом совершала операцию перехода в зависимости от того, положительное оно или отрицательное. Если в считывающее устройство вводились перфокарты с другой программой, то машина исполняла другой набор операций. Аналитическая машина была программируемой, поэтому ей было необходимо программное обеспечение. Чтобы создать это программное обеспечение, Бэббидж был вынужден нанять молодую женщину – Аду Августу Лавлейс (Ada Augusta Lovelace), дочь знаменитого британского поэта Байрона. Таким образом, Ада Лавлейс стала первым в мире программистом. Впоследствии в ее честь был назван один из языков программирования – Ada

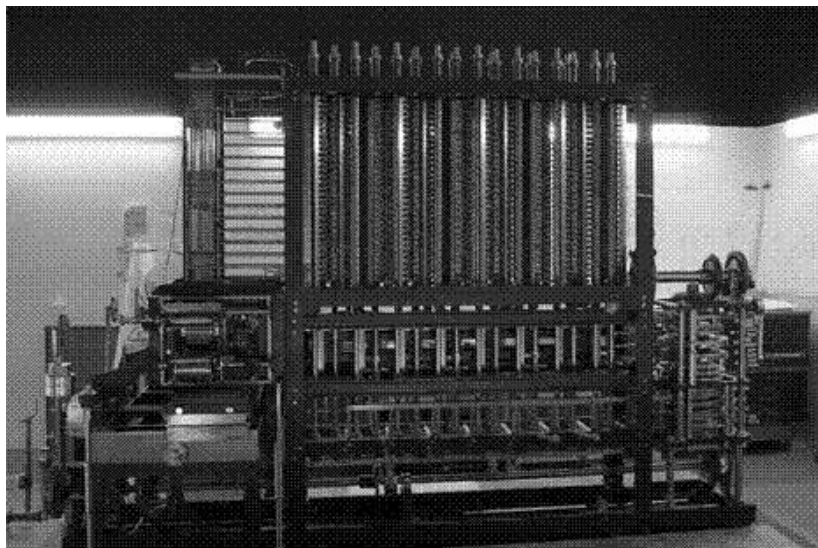


Рисунок 2.2 – Машина Бэббиджа

Идеи Бэббиджа опередили его эпоху. Сегодня большинство современных компьютеров конструктивно сходны с аналитической машиной. Развитием счетных машин занимались и в Америке. В конце 30-х годов XX века этим занялись два американца: Джон Атанасов (John Atanasoff) из Колледжа штата Айова и Джордж Стибиц из Bell Labs. Атанасовым была разработана чрезвычайно для своего времени мощная машина. В ней использовалась двоичная арифметика и память на базе конденсаторов. Конденсаторы периодически обновлялись, чтобы избежать утечки заряда. Современная динамическая память (ОЗУ) работает точно по такому же принципу.

2.1.2 Первое поколение – электронные лампы (1945-1955)

Вторая мировая война стала мощным стимулом к разработке электронного компьютера. Германские адмиралы посылали закодированные с помощью прибора под названием ENIGMA команды на свои подводные лодки. Англичане перехватывали эти команды, но расшифровать их не могли.

В начале войны англичане смогли приобрести у поляков этот прибор ENIGMA. Однако, чтобы расшифровать закодированное послание, требовалось огромное количество вычислений, и их нужно было произвести сразу после перехвата радиограммы. Для этого британское правительство основало секретную лабораторию, в которой был создан электронный компьютер под названием COLOSSUS (рисунок 2.3). В создании этой машины принимал участие знаменитый британский математик Алан Тьюринг. COLOSSUS был создан в 1943 году. Это был первый в мире электронный компьютер.



Рисунок 2.3 – Машина COLOSSUS

Немного позже ученый Джон Моушли, который был знаком с работами Атанасова и Стиббца, обратился к американским властям с заявкой на финансирование работ по созданию электронного компьютера. Заявка была удовлетворена в 1943 году, после чего Моушли со своим студентом Дж. Преспером Экертом (J. Presper Eckert) начали конструировать свой электронный компьютер. Они назвали его ENIAC (Electronic Numerical Integrator and Computer – электронный цифровой интегратор и калькулятор). Создавали его они до 1946 года. Но тогда он уже был не нужен – по крайней мере, для достижения первоначально поставленных целей. После окончания войны Моушли и Экерт организовали школу, где они рассказывали о своем компьютере коллегам-ученым. В этой школе зародился интерес к созданию больших цифровых компьютеров.

Один из участников проекта ENIAC, Джон фон Нейман, в Институте специальных исследований в Принстоне создал собственную электронную машину под названием IAS (Immediate Address Storage – память с прямой адресацией) (рисунок 2.4). Он первый осознал, что программирование компьютеров с большим количеством переключателей и кабелей – занятие очень медленное, неудобное и утомительное. Именно он придумал представлять программу в памяти компьютера в цифровой форме вместе с данными, а также заменил двоичной арифметикой десятичную, которая использовалась в машине ENIAC.

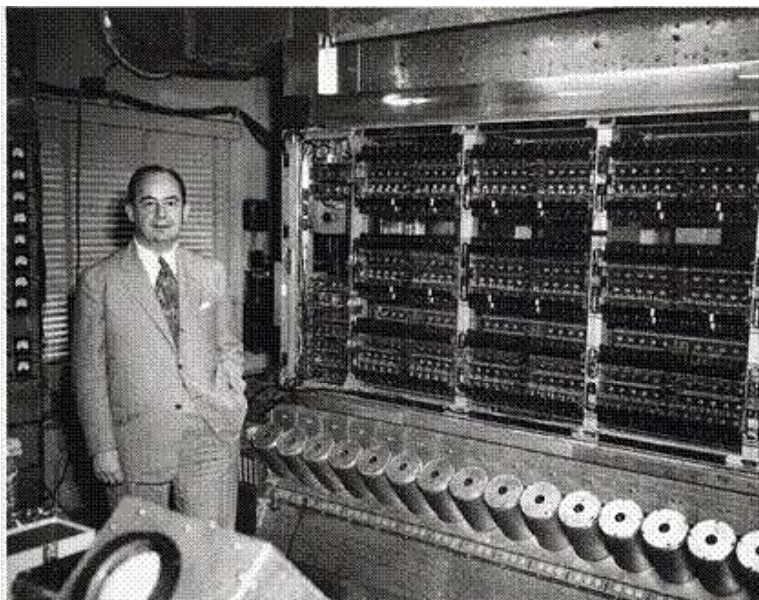


Рисунок 2.4 – Машина IAS

Так была изобретена машина, которая носит название фон-неймановская вычислительная машина. Идея данной машины была использована в EDSAC, первой машине с программой в памяти. И спустя более чем полвека идеи фон Неймана являются основой большинства современных цифровых компьютеров. Архитектура этой машины представлена на рисунке 2.5.

Примерно в то же время, когда фон Нейман работал над машиной IAS, исследователи Массачусетского технологического института (МТИ) разрабатывали свой компьютер Whirlwind I. В отличие от IAS, ENIAC и других машин того же типа со словами большой длины, предназначенными для серьезных вычислений, машина Whirlwind I имела слова по 16 бит и предназначалась для работы в реальном времени. Этот проект привел к изобретению Джеймсом Форрестером (Jay Forrester) памяти на магнитном сердечнике, а затем и первого серийного мини-компьютера.

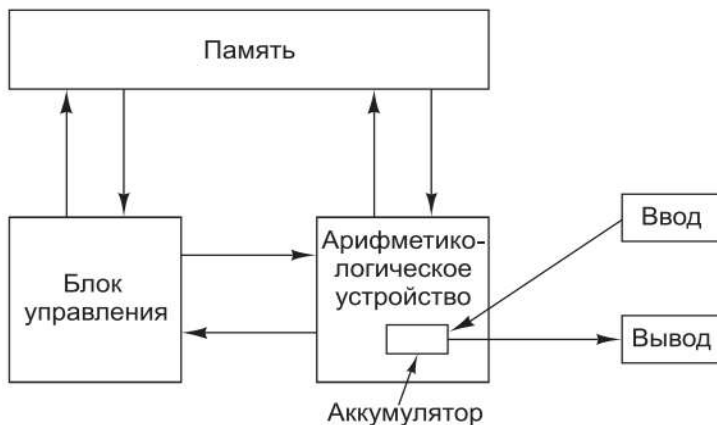


Рисунок 2.5 – Схема архитектуры фон-неймановской вычислительной машины

2.1.3 Второе поколение – транзисторы (1955-1965)

В 1947 сотрудниками лаборатории Bell Laboratories Джоном Бардином (John Bardeen), Уолтером Браттейном (Walter Brattain) и Уильямом Шокли (William Shockley) был разработан электронный усилитель – транзистор. В течение десяти лет транзисторы произвели революцию в производстве компьютеров, и к концу 50-х годов компьютеры на вакуумных лампах стали пережитком прошлого. Первый компьютер на транзисторах был построен в лаборатории МТИ. Его называли TX-0 (Transistorized eXperimental computer 0 – экспериментальная транзисторная вычислительная машина 0). Он был предназначен только для тестирования будущей машины TX-2. Но позже оказалось, что машина TX-2 не произвела большого значения.

Кеннет Ольсен (Kenneth Olsen), инженер лаборатории МТИ, в 1957 году основал компанию DEC (Digital Equipment Corporation – корпорация по производству цифровой аппаратуры). Компания начала производить свою машину, которая была схожа с TX-0. Эта машина, которую они впоследствии назвали PDP-1, появилась только в 1961 году. Причиной столь позднего выхода машины является, главным образом то, что финансисты компании DEC считали, что у производства компьютеров нет будущего.

Компьютер PDP-1 имел память, состоящую из 4096 слов по 18 бит и мог выполнить 200 000 команд в секунду. Компьютер PDP-1 обладал большой стоимостью, а именно 120 000 долларов. Компания DEC продала десятки компьютеров, и дала начало в развитии компьютерной промышленности.

Через несколько лет компания DEC разработала модель PDP-8 (12-разрядный компьютер). Стоимость PDP-8 была гораздо дешевле стоимости PDP-1 – 16000 долларов. Главным отличием PDP-8 от PDP-1 было в том, что в PDP-8

стали использовать специальное устройство, называемое шина (omnibus) (рисунок 2.6). Шина – это набор параллельно соединенных проводов, связывающих компоненты компьютера. Это нововведение радикально отличало PDP-8 от IAS. Было продано 50000 компьютеров модели PDP-8, что позволило ей стать лидером на рынке мини-компьютеров.



Рисунок 2.6 – Шина

В 1964 году небольшая малоизвестная компания CDC (Control Data Corporation) выпустила машину 6600. Эта машина работала почти на порядок быстрее, чем все остальные машины того времени (рисунок 2.7).



Рисунок 2.7 – Машина 6600

В связи с тем, что компьютер 6600 для сложных расчетов пользовался большой популярностью, бизнес компании CDC пошел «в гору». Секрет столь высокого быстродействия заключался в том, что внутри центрального процессора находилась машина с высокой степенью параллелизма. У нее было несколько функциональных устройств для сложения, умножения и деления, при этом все они могли работать одновременно. Быстрая работа машины требовала тщательной работы программиста, но при определенных усилиях можно было сделать так, чтобы машина исполняла 10 команд одновременно.

Внутри машины 6600 было встроено несколько маленьких компьютеров.

Таким образом, центральный процессор занимался только вычислениями, а остальные функции (управление работой машины, а также ввод и вывод информации) были наложены на эти маленькие компьютеры. Компьютер 6600 на десятилетия опередил свое время. Многие ключевые принципы современных компьютеров уходят корнями к 6600.

2.1.4 Третье поколение – интегральные схемы (1965-1980)

В 1958 году Джек Килби (Jack Kilby) и Роберт Нойс (Robert Noyce) изобрели кремниевую интегральную схему. Это позволило размещать на одной небольшой микросхеме десятки транзисторов. Компьютеры на интегральных схемах имели большое преимущество над их предшественниками: они резко уменьшились в размерах, при этом работали быстрее. А также уменьшилась их стоимость.

Лидер компьютерного рынка на тот момент, компания IBM, выпустила линейку транзисторных компьютеров System/360 (рисунок 2.8), которые были предназначены как для научных, так и для коммерческих расчетов. Линейка System/360 имела много нововведений. Это было семейство компьютеров, которое работало с языком, называемым ассемблер. В данное семейство входили машины разного размера и вычислительной мощности. Идея создания семейств компьютеров вскоре стала очень популярной, и в течение нескольких лет большинство компьютерных компаний выпустили серии сходных машин с разной стоимостью и функциями.



Рисунок 2.8 – System/360

Еще одно важное нововведение в System/360 – многозадачность. В памяти компьютера могло находиться одновременно несколько программ, и пока одна программа ждала, когда закончится процесс ввода-вывода, другая могла исполняться. Благодаря этому ресурсы процессора расходовались намного лучше, чем у его предшественников. Компьютер System/360 был первой машиной, которая могла полностью эмулировать (имитировать) работу других компьютеров.

2.1.5 Четвертое поколение – сверхбольшие интегральные схемы (1980)

Четвертое поколение – это время появления *сверхбольших интегральных схем (СБИС)*. Их появление позволило размещать на одной плате десятки, сотни тысяч, а впоследствии и миллионы транзисторов. Это привело к созданию компьютеров маленьких размеров и более быстродействующих. До появления PDP-1 компьютеры были очень большие и дорогостоящие. Компаниям и университетам приходилось иметь целые вычислительные центры. К 80-м годам цены на компьютеры упали. Это дало возможность приобретать компьютеры не только организациям, но и отдельным людям. Так началась эпоха персональных компьютеров.

Применение персональных компьютеров было направлено для обработки текстов, электронных таблиц, а также для исполнения приложений с высоким уровнем интерактивности (например, игр), для которых большие компьютеры не подходили.

Первые персональные компьютеры продавались, как правило, в виде сборочных комплектов. Состав комплекта был следующий: набор интегральных схем, печатная плата, источник питания, несколько кабелей и иногда дисконвод. Основным недостатком этих компьютеров являлось то, что программное обеспечение к компьютеру не прилагалось. Покупателю приходилось писать программное обеспечение самому. Позднее появилась операционная система CP/M, написанная Гари Килдаллом (Gary Kildall) для Intel 8080. Это была полноценная операционная система (на дискете), со своей файловой системой и интерпретатором для исполнения пользовательских команд, которые вводились с клавиатуры.

Наблюдая за разработками различных компьютерных компаний, IBM – лидер компьютерного рынка, тоже решила заняться производством персональных компьютеров. Сотрудник IBM Филипп Эстридж (Philip Estridge), взял за основу процессор Intel 8088 и построил персональный компьютер из разнородных компонентов. Этот компьютер (IBM PC) появился в 1981 году и стал самым покупаемым компьютером в истории. Потенциал производства IBM PC был настолько велик, что создавало огромные трудности в развитии других компаний.

В 1984 году компания Apple выпустила компьютер Apple Macintosh. Он являлся наследником компьютера компании Apple Lisa – первого компьютера с графическим интерфейсом.

Первая версия IBM PC была оснащена операционной системой MS-DOS, которую разработала тогда еще крошечная корпорация Microsoft. Компания Intel выпускала все более мощные процессоры. Этим воспользовались компании IBM и Microsoft. Они совместно разработали операционную систему OS/2, отличительной особенностью которой был *графический пользовательский интерфейс* (Graphical User Interface, GUI), сходный с интерфейсом Apple Macintosh. Параллельно с этим компания Microsoft также разрабатывала собственную операционную систему Windows, которая работала на основе MS-DOS, – на случай, если OS/2 не будет иметь спроса. Позже оказалось, что OS/2 действительно не стала пользоваться спросом. Таким образом, Microsoft продолжила развивать и выпускать операционную систему Windows.

Первоначальный успех процессора 8088 воодушевил компанию Intel на его дальнейшее усовершенствование. Особо примечательна версия 80386, выпущенная в 1985 году. Это был уже 32-разрядный процессор. Далее последовал улучшенный вариант этого процессора – процессор 80486. Последующие версии назывались Pentium и Core. Эти микросхемы используются практически во всех современных PC. Архитектура этих процессоров обозначается как x86. Совместимые микросхемы, производимые фирмой AMD, тоже называются x86.

Вплоть до 1992 года персональные компьютеры были 8-разрядными, 16-разрядными и 32-разрядными. Затем появилась революционная 64-разрядная модель Alpha производства DEC, намного превосшедшая по показателям производительности все прочие ПК. Впрочем, тогда коммерческий успех этой модели оказался весьма скромным – лишь через десятилетие 64-разрядные машины приобрели популярность.

2.1.6 Пятое поколение – компьютеры небольшой мощности и невидимые компьютеры

В 1981 году правительство Японии объявило о намерениях выделить национальным компаниям 500 миллионов долларов на разработку компьютеров пятого поколения на основе технологий искусственного интеллекта, которые должны были потеснить, по их мнению, машины четвертого поколения. Наблюдая за тем, как японские компании оперативно захватывают рыночные позиции в самых разных областях промышленности – от фотоаппаратов до стереосистем и телевизоров, – американские и европейские производители начали требовать у своих правительств аналогичное финансирование. Однако японский проект разработки компьютеров пятого поколения в конечном итоге показал свою несостоятельность и был тихо свернут. В каком-то смысле эта ситуация оказалась близка той, с которой столкнулся Беббидж, – идея настолько опередила свое время, что для ее реализации не нашлось адекватной технологической базы.

Тем не менее то, что можно назвать пятым поколением компьютеров, все же материализовалось. Дело в том, что компьютеры начали стремительно

уменьшаться. В 1989 году фирма Grid Systems выпустила первый планшетный компьютер, который назывался GridPad (рисунок 2.9).



Рисунок 2.9 – GridPad

Он был оснащен небольшим экраном, на котором пользователь мог писать специальным пером. Плюсами данной системы является, то что пользователь может носить планшетный компьютер с собой. А также сенсорный экран и распознавание рукописного текста дает легкость его использования. В 1993 году появляется модель Apple Newton. Данная модель наглядно доказала, что компьютер можно уместить в корпусе размером с кассетный плеер. Как и GridPad, Newton использовал рукописный ввод.

Планшетные компьютеры изменялись и эволюционировали. Так появились знакомые нам *смартфоны*. В данный момент существуют несколько платформ. Наиболее популярными из них являются платформы Apple iPhone и Google Android.

2.2 МЕТОДЫ КЛАССИФИКАЦИИ КОМПЬЮТЕРОВ

Существует достаточно много систем классификации компьютеров. Мы рассмотрим лишь некоторые из них, выбрав те, о которых наиболее часто упоминают в доступной технической литературе и средствах массовой информации.

2.2.1 Классификация по назначению

Одним из наиболее ранних способов классификации является классификация по назначению. Этот способ связан с тем, как компьютер применяется. По этому принципу можно различить:

большие ЭВМ (электронно-вычислительные машины);

мини-ЭВМ;

микро-ЭВМ;

персональные компьютеры.

Персональные компьютеры подразделяют на:

массовые;

деловые;

портативные;
развлекательные;
рабочие станции.

Рассмотрим каждую группу назначения.

Большие ЭВМ. Это самые мощные компьютеры. Их применяют для обслуживания очень крупных организаций и даже целых отраслей народного хозяйства. За рубежом компьютеры этого класса называют *мэйнфреймами (mainframe)*. В России за ними закрепился термин *большие ЭВМ*. Штат обслуживания большой ЭВМ достигает десятков человек. На базе таких суперкомпьютеров создают *вычислительные центры*, включающие в себя несколько отделов или групп.

Мини-ЭВМ. От больших ЭВМ компьютеры этой группы отличаются уменьшенными размерами и, соответственно, меньшей производительностью и стоимостью. Такие компьютеры используются крупными предприятиями, научными учреждениями и некоторыми высшими учебными заведениями, сочетающими учебную деятельность с научной.

Микро-ЭВМ. Компьютеры данного класса доступны многим потребителям. Организации, использующие микро-ЭВМ, обычно не создают вычислительные центры. Для обслуживания такого компьютера им достаточно небольшой вычислительной лаборатории в составе нескольких человек. В число сотрудников вычислительной лаборатории обязательно входят программисты, хотя напрямую разработкой программ они не занимаются. Необходимые системные программы обычно покупают вместе с микро-ЭВМ, а разработку нужных прикладных программ заказывают более крупным вычислительным центрам или специализированным организациям.

Персональные компьютеры. Эта категория компьютеров получила особое бурное развитие в течение последних тридцати лет. Такой компьютер предназначен для обслуживания одного рабочего места. Как правило, с персональным компьютером (ПК) работает один человек. Современные ПК обладают немалой производительностью, несмотря на свои небольшие размеры и относительно невысокую стоимость. Персональный компьютер (*Personal Computer, PC*) вполне способен удовлетворить большинство запросов обычного пользователя.

2.2.2 Классификация по уровню специализации

По уровню специализации компьютеры делят на *универсальные* и *специализированные*. На базе универсальных компьютеров можно собирать вычислительные системы произвольного состава (состав компьютерной системы называется *конфигурацией*). Так, например, один и тот же ПК можно использовать для работы с текстами, музыкой, графикой, фото и видеоматериалами.

Специализированные компьютеры предназначены для решения определенного круга задач. К таким компьютерам можно отнести, например, бортовые компьютеры автомобилей, судов, самолетов, космических аппаратов.

Бортовые компьютеры управляют средствами ориентации и навигации, осуществляют контроль состояния бортовых систем, выполняют некоторые функции автоматического управления и связи, а также большинство функций по оптимизации параметров работы систем объекта (например, оптимизацию расхода топлива в зависимости от конкретных условий движения объекта).

Специализированные мини-ЭВМ, ориентированные на работу с графикой, называют *графическими станциями*. Их используют при подготовке кино- и видеофильмов, а также рекламной продукции.

Специализированные компьютеры, объединяющие компьютеры предприятия в одну сеть, называют *файловыми серверами*.

Компьютеры, обеспечивающие передачу информации между различными участниками всемирной компьютерной сети, называют *сетевыми серверами*.

2.2.3 Классификация по типоразмерам

Следующий тип классификации – по типоразмеру. Различают *портативные (notebook)*, *карманные (palmtop)* и *настольные (desktop)* модели.

Наиболее широко распространены *настольные модели*. Они являются принадлежностью рабочего места. Эти модели отличаются простотой изменения конфигурации за счет несложного подключения дополнительных внешних приборов или установки дополнительных внутренних компонентов. Еще одной особенностью является то, что достаточные размеры корпуса в настольном исполнении позволяют выполнять большинство подобных работ без привлечения специалистов. Это позволяет настраивать компьютерную систему оптимально для решения именно тех задач, для которых она была приобретена.

По транспортировке более удобны *портативные модели*. Благодаря сближению стоимости портативных и настольных моделей сегодня их применяют буквально все категории пользователей – от руководителей предприятий и организаций до студентов и школьников. С портативным компьютером можно работать при отсутствии рабочего места. Особая привлекательность портативных компьютеров связана с тем, что их можно использовать в качестве средства связи. Портативный компьютер можно подключить к мобильному телефону, к обычной телефонной сети или к беспроводной компьютерной сети. Это дает возможность выхода в Интернет практически из любой географической точки. Но при этом для эксплуатации на рабочем месте портативные компьютеры не очень удобны. Исключением может быть, если их подключить к настольным компьютерам, используемым стационарно.

Карманные модели выполняют функции «интеллектуальных записных книжек». Они позволяют хранить оперативные данные и получать к ним быстрый доступ. Некоторые карманные модели имеют жестко встроенное программное обеспечение, что облегчает непосредственную работу, но снижает гибкость в выборе прикладных программ.

2.2.4 Классификация по совместимости

Последней рассматриваемой нами классификацией является классификация по совместимости.

В мире существует множество различных видов и типов компьютеров. Они выпускаются разными производителями, собираются из разных деталей, работают с разными программами. При этом очень важным вопросом становится *совместимость* различных компьютеров между собой. От совместимости зависит взаимозаменяемость узлов и приборов, предназначенных для разных компьютеров, возможность переноса программ с одного компьютера на другой и возможность совместной работы разных типов компьютеров с одними и теми же данными.

Различают следующие типы совместимости:

- *аппаратная совместимость*;
- *совместимость на уровне операционной системы*;
- *программная совместимость*;
- *совместимость на уровне данных*.

По аппаратной совместимости различают так называемые *аппаратные платформы*. В области персональных компьютеров сегодня наиболее широко распространены две аппаратные платформы – *IBM PC* и *Apple Macintosh*. Кроме них существуют и другие платформы, распространенность которых ограничивается отдельными регионами или отдельными отраслями. Принадлежность компьютеров к одной аппаратной платформе повышает совместимость между ними, а принадлежность к разным платформам – понижает.

2.3 АППАРАТНАЯ КОНФИГУРАЦИЯ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

2.3.1 Базовая конфигурация персонального компьютера

Персональный компьютер – универсальная техническая система. Его *конфигурацию* (состав оборудования) можно гибко изменять по мере необходимости. Тем не менее существует понятие *базовой конфигурации*, которую считают типовой. Понятие базовой конфигурации может меняться. В настоящее время в базовой конфигурации рассматривают четыре устройства:

- системный блок;
- монитор;
- клавиатура;
- мышь.

Системный блок представляет собой основной узел, внутри которого расположены другие важные компоненты (рисунок 2.10).

Устройства, которые находятся внутри системного блока, называют *внутренними*, а устройства, подключаемые к системному блоку снаружи, – *внешними*. Внешние устройства, предназначенные для ввода, вывода и длительного

хранения данных, также называются *периферийными*.



Рисунок 2.10 – Системный блок

По внешнему виду системные блоки различаются формой корпуса. Обычно персональные компьютеры выпускают в горизонтальном (desktop) или вертикальном (tower) корпусе. Корпуса в вертикальном исполнении различаются габаритами: *полноразмерный* (big tower), *среднеразмерный* (midi tower) и *малоразмерный* (mini tower). Среди корпусов, имеющих горизонтальное исполнение, выделяют *плоские* и *особо плоские* (slim).

Кроме формы для корпуса важен *форм-фактор* – это параметр, от которого зависят требования к размещаемым устройствам. Морально устаревшим стандартом корпуса для персональных компьютеров является форм-фактор AT. В настоящее время основным используемым форм-фактором корпуса является ATX. Форм-фактор корпуса необходимо согласовывать с форм-фактором системной (*материнской*) платы компьютера.

Вместе с корпусом персонального компьютера обычно поставляется блок питания, и, таким образом, мощность блока питания также является одним из параметров корпуса. Для массовых моделей достаточной можно считать мощность блока питания 300 – 450 Вт.

Монитор – устройство для визуального представления данных (рисунок 2.11). Это одно из главных устройств, предназначенных для вывода информации. К основным параметрам монитора относятся: тип, размер и шаг маски экрана, максимальная частота регенерации изображения, класс защиты.

В настоящее время наибольшее распространение получили плоские жидкокристаллические мониторы (ЖК). Они пришли на смену мониторам, основанным на электронно-лучевой трубке (ЭЛТ).



Рисунок 2.11 – Монитор

Размер монитора измеряется между противоположными углами видимой части экрана по диагонали. Единицей измерения является дюйм.

Частота регенерации (обновления) изображения показывает, сколько раз в течение одной секунды монитор может сменить изображение (другое название – *частота кадров*). Величина этого параметра зависит не только от характеристик монитора, но и от свойств видеоадаптера.

Частоту регенерации изображения измеряют в герцах (Гц). Чем выше частота, тем получаемое изображение четче и устойчивее, вызывает меньшее утомление глаз, позволяет больше времени работать с компьютером. При частоте регенерации около 60 Гц мелкое мерцание изображения обычно замечают даже невооруженным глазом. Сегодня такое значение считается недопустимым для практического использования. Для ЭЛТ-мониторов минимально допустимым значением считают 75 Гц, нормальным – 85 Гц и комфортным – 100 Гц и более. У жидкокристаллических мониторов изображение обладает большей инерционностью, так что мерцание подавляется автоматически. Для них частота обновления в 75 Гц уже считается комфортной.

Класс защиты монитора определен тем стандартом, которому соответствует монитор с точки зрения требований техники безопасности.

Клавиатура – одно из основных устройств управления персональным компьютером (рисунок 2.12). Служит для ввода *алфавитно-цифровых* (знаковых) данных, а также подачи команд управления. Комбинация монитора и клавиатуры обеспечивает простейший *интерфейс пользователя*. Клавиатура используется для управления компьютерной системой, а монитор позволяет получить от нее отклик.

Стандартная клавиатура имеет более 100 клавиш, которые функционально распределены на несколько групп:

группа алфавитно-цифровых клавиш предназначена для ввода знаковой

информации и команд, набираемых по буквам. Каждая клавиша может работать в нескольких режимах (*регистрах*) и, соответственно, может использоваться для ввода нескольких символов. Удержание клавиши SHIFT (нефиксированное переключение) обеспечивает переключение между *нижним регистром* (для ввода строчных символов) и *верхним регистром* (для ввода прописных символов). Если возникла необходимость переключить регистр на продолжительное время используется клавиша CAPS LOCK (фиксированное переключение);

группа функциональных клавиш включает двенадцать клавиш (от F1 до F12), размещенных в верхней части клавиатуры. Функции, которые реализуют эти клавиши, зависят как от свойств работающей в данный момент программы, так и от свойств операционной системы. Общепринято соглашение, что клавиша F1 должна вызывать справочную систему, в которой можно найти информацию о действии остальных клавиш и их комбинаций;

служебные клавиши располагаются рядом с клавишами алфавитно-цифровой группы. Так как ими пользуются особенно часто, то они имеют увеличенный размер. К ним относятся клавиши SHIFT и ENTER, регистровые клавиши ALT и CTRL (их используют в комбинации с другими клавишами для формирования команд), клавиша TAB (для ввода позиций табуляции при наборе текста), клавиша ESC (для отказа от исполнения начатой операции), и клавиша BACKSPACE (для удаления только что введенных знаков);

группа клавиш управления курсором расположена справа от алфавитно-цифровой панели. Клавиши управления курсором позволяют управлять позицией ввода, а сам курсор используется при работе с программами, выполняющими ввод данных и команд с клавиатуры. Четыре клавиши со стрелками выполняют смещение курсора в направлении, указанном стрелкой;

группа клавиш дополнительной панели позволяет продублировать действие цифровых и знаковых клавиш основной панели. Использование этой группы клавиш регулируется включением/отключением клавиши NUM LOCK (о состоянии переключателя NUM LOCK можно судить по светодиодным индикаторам, расположенным в правом верхнем углу клавиатуры).



Рисунок 2.12 – Клавиатура

Мышь – устройство управления манипуляторного типа. Наибольшее распространение сегодня получили мыши, с двумя управляющими кнопками и вращающимся колесиком-регулятором в качестве еще одного элемента управления (рисунок 2.13).



Рисунок 2.13 – Мышь

Функции органов управления определяются тем программным обеспечением, которое поставляется вместе с устройством.

Управление компьютером осуществляется с помощью перемещения мыши и кратковременным нажатием (щелчком) на правую или левую кнопку. Перемещение мыши по плоской поверхности синхронизируется с перемещением графического объекта (*указателя мыши*) на экране монитора. В отличие от клавиатуры мышь не используется для ввода знаковой информации – принцип ее использования является *событийным*. Программа-драйвер, управляющая работой мыши, отслеживает перемещение мыши, а щелчок по ее кнопкам является *событием*. Анализируя эти события, драйвер устанавливает время, когда произошло событие и в каком месте экрана в этот момент находился указатель. Эти данные передаются в прикладную программу, с которой работает пользователь в данный момент. По этим данным программа может определить команду, которую хотел выполнить пользователь, и приступить к ее исполнению.

Монитор и мышь совместно обеспечивают современный тип интерфейса пользователя, который называется *графическим*. Пользователь наблюдает на экране графические *объекты* и *элементы управления*. *Свойства объектов* могут быть им изменены с помощью мыши, при этом приводятся в действие *элементы управления* компьютерной системой, а с помощью монитора получается отклик от нее в графическом виде.

Параметры мыши, которые допускают регулировку пользователями, являются: *чувствительность* (определяет на какую величину необходимо переместить указатель на экране при совершенном пользователем линейном перемещении мыши), функции левой и правой кнопок, а также *чувствительность к двойному нажатию* (определяет интервал времени, при котором двойное нажатие на кнопку мыши будет расцениваться как один щелчок).

2.3.2. Внутренние устройства системного блока

Материнская плата – основная плата персонального компьютера (рисунок 2.14). На ней размещаются:

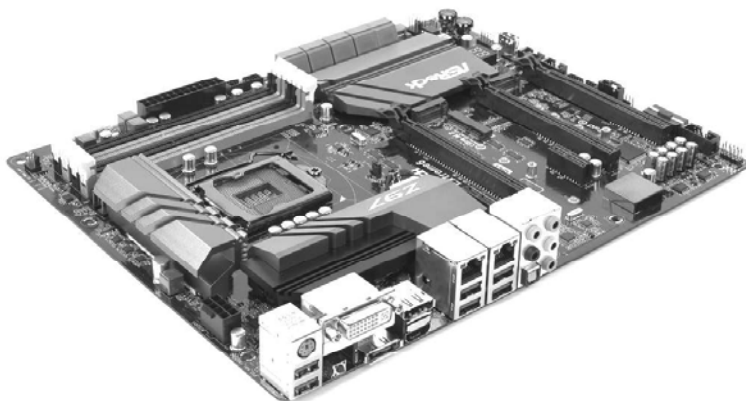


Рисунок 2.14 – Материнская плата

- процессор – основная микросхема, выполняющая большинство математических и логических операций;

- микропроцессорный комплект (чипсет) – набор микросхем, управляющих работой внутренних устройств компьютера и определяющих основные функциональные возможности материнской платы;

- шины – наборы проводников, по которым происходит обмен сигналами между внутренними устройствами компьютера;

- оперативная память (оперативное запоминающее устройство, ОЗУ) – набор микросхем, предназначенных для временного хранения данных, когда компьютер включен;

- ПЗУ (постоянное запоминающее устройство) – микросхема, предназначенная для длительного хранения данных, в том числе и когда компьютер выключен;

- разъемы для подключения дополнительных устройств (слоты).

Жесткий диск – устройство, предназначенное для организации хранения больших объемов данных и программ в течение длительного времени (рисунок 2.15).



Рисунок 2.15 – Жесткий диск

К основным параметрам жестких дисков относятся *емкость* и *производительность*.

Конструктивно жесткий диск – это группа дисков со специальным магнитным покрытием, имеющих общую ось вращения, и вращающихся с высокой скоростью. Таким образом, современный жесткий диск имеет не две поверхности, как это было бы у обычного плоского диска, а $2n$ поверхностей, где n – число отдельных дисков в группе.

Большинство фирм, производящих жесткие диски, применяют в своих устройствах технологию *гигантского магниторезистивного эффекта* (GMR – Giant Magnetic Resistance). Над рабочей поверхностью диска располагают головку, предназначенную для чтения-записи данных. При высоких скоростях вращения дисков (90-250 об/с) в зазоре между поверхностью диска и головкой образуется аэродинамическая подушка, и головка движется над магнитной поверхностью на высоте, составляющей несколько микрон. Для записи данных через головку пропускают ток, при изменении величины которого происходит изменение напряженности магнитного поля в зазоре, что приводит к изменению магнитного поля ферромагнитных частиц, образующих покрытие диска. Операция считывания данных происходит в обратном порядке. Частицы покрытия, пронесшиеся на высокой скорости вблизи головки, наводят в ней ЭДС самоиндукции. Электромагнитные сигналы, возникающие при этом, усиливаются и передаются на обработку.

Для управления работой жесткого диска используется специальное аппаратно-логическое устройство – *контроллер жесткого диска*. В современных компьютерах функции контроллеров диска частично интегрированы в сам жесткий диск, а частично выполняются микросхемами, входящими в микропроцессорный комплект (чипсет).

Практически все современные жесткие диски обладают очень высоким показателем скорости внутренней передачи данных (до 40 – 100 Мбайт/с), и потому их производительность в первую очередь определяется характери-

ками интерфейса, с помощью которого они связаны с материнской платой.

Кроме скорости передачи данных с производительностью диска связан параметр *среднего времени доступа*. Это такой интервал времени, который необходим для поиска нужных данных, и зависит от скорости вращения диска. Для дисков, вращающихся с частотой 5400 об/мин, среднее время доступа к данным составляет 9 – 10 мкс, для дисков с частотой 7200 об/мин – 7 – 8 мкс. Изделия более высокого уровня обеспечивают среднее время доступа к данным 4 – 6 мкс.

Дисководы оптических дисков. В конце XX века в базовую конфигурацию компьютера стали включать дисководы CD-ROM, позднее – DVD-ROM. Аббревиатура CD-ROM (Compact Disc Read-Only Memory) переводится на русский язык как *постоянное запоминающее устройство на основе компакт-диска*. Работа этих устройств основывалась на принципе считывания данных с помощью лазерного луча, отражающегося от поверхности диска. Запись данных на компакт-диске отличалась от записи на магнитных дисках очень высокой плотностью. Стандартный компакт-диск хранил примерно 650 Мбайт данных.

Все современные дисководы для работы с оптическими носителями поддерживают работу с так называемыми DVD-дисками (рисунок 2.16).



Рисунок 2.16 – DVD-дисковод

Стандартный DVD-диск хранит до 4,7 Гбайт информации. Существуют и более емкие оптические носители информации, например, Blu-Ray Disc, емкость которых доходит до нескольких сотен гигабайт. Компьютер, оснащенный современным оптическим дисководом, обычно поддерживает и носители предыдущих поколений. Современные диски имеют те же размеры, что и CD, но плотность записи на них гораздо выше, нередко для хранения данных используют несколько слоев и две стороны диска.

Большие объемы данных характерны для *мультимедийной информации* (видео, графика, музыка), поэтому CD/DVD/BD-дисководы относят к аппаратным средствам мультимедиа.

Основным недостатком стандартных оптических дисководов первоначально являлась невозможность записи данных, но сегодня все устройства имеют функцию записи оптических дисков – дисководы CD/DVD/BD-RW. Для записи используются специальные заготовки. Некоторые из них допускают однократную запись (после записи диск превращается в обычный оптический диск, доступный только для чтения), другие позволяют стереть ранее записанную информацию и выполнить запись заново.

Основным параметром оптических дисководов является скорость чтения данных. Она измеряется в кратных долях. За единицу измерения принята скорость чтения музыкальных компакт-дисков, составляющая в пересчете на данные 150 Кбайт/с. Таким образом, CD-ROM-дисковод с удвоенной скоростью чтения обеспечивает производительность 300 Кбайт/с, с учетверенной скоростью – 600 Кбайт/с и т. д. В настоящее время наибольшее распространение имеют устройства чтения CD-ROM с производительностью $48\times$ – $56\times$. Для заготовок, рассчитанных на однократную запись, скорость записи в соответствующих устройствах не уступает скорости чтения. Для заготовок многократной записи скорость записи может составлять $12\times$ – $24\times$.

Дисководы для работы с DVD- и BD-дисками обеспечивают более высокие скорости передачи данных при тех же скоростях вращения диска за счет более высокой плотности носителей.

Так, единица измерения скорости для DVD-диска составляет около 1352 Кбайт/с, то есть 1,32 Мбайт/с. Для BD – дисков тот же показатель равен примерно 4,5 Мбайт/с.

Видеокарта (рисунок 2.17) совместно с монитором создают *видеоподсистему* персонального компьютера.

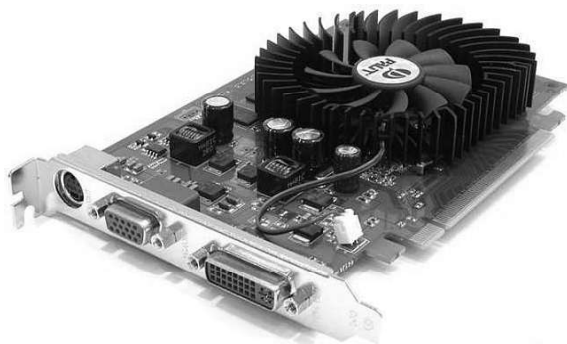


Рисунок 2.17 – Видеокарта

На начальном этапе развития вычислительной техники видеокарта не входила в состав основных компонентов ПК. В оперативной памяти выделяли специальную *область памяти*, в которую процессор заносил данные об изображении и использовал их для работы с монитором. Специальный *контроллер*

экрана считывал данные о яркости отдельных точек экрана из ячеек памяти этой области и в соответствии с этими данными управлял разверткой горизонтального луча электронной пушки монитора.

С переходом от монохромных (черно-белых) мониторов к цветным и с увеличением *разрешения экрана* (количества точек по вертикали и горизонтали) выделяемой области видеопамати перестало хватать для хранения графических данных, а процессор перестал успевать производить построение и обновление изображений. В это время и произошло выделение всех операций, связанных с управлением экраном, в отдельный блок операций, и их выполнение было поручено *видеоадаптеру*. Физически видеоадаптер выполнен в виде отдельной *дочерней платы*, которая вставляется в один из слотов материнской платы и называется *видеокартой*. Видеоадаптер взял на себя функции *видеоконтроллера, видеопроцессора и видеопамати*.

За время существования персональных компьютеров сменилось несколько стандартов видеоадаптеров: MDA (монохромный); CGA (4 цвета); EGA (16 цветов); VGA (256 цветов). В настоящее время применяются видеоадаптеры SVGA, обеспечивающие воспроизведение до 16,7 миллиона цветов с возможностью произвольного выбора разрешения экрана из стандартного ряда значений (640×480 , 800×600 , 1024×768 , 1152×864 ; 1280×1024 точек и далее).

Разрешение экрана также является одним из важнейших параметров видеоподсистемы. Чем выше разрешение экрана, тем больше информации можно на нем отобразить, но размер каждой отдельной точки при этом уменьшается и соответственно уменьшается видимый размер элементов изображения. Использование высокого разрешения на мониторах малых размеров приводит к тому, что элементы изображения становятся неразборчивыми, и работа с документами и программами вызывает сильное утомление органов зрения. Использование низкого разрешения приводит к тому, что элементы изображения становятся крупными, но на экране их располагается очень мало. Это сказывается на производительности труда и приводит к неэффективной работе. Таким образом, для каждого размера монитора нужно подбирать свое оптимальное разрешение экрана, которое и должен обеспечить видеоадаптер.

Для нормальной работы с документами, оформленными с размером, соответствующими стандартным листам бумаги формата A4, как правило, необходимо экранное разрешение не менее 1024×768 и размер монитора не менее 17 дюймов. Для большинства прикладных программ, в настоящее время, оптимальным также является разрешение 1024×768 и более, хотя в случае необходимости программы, как правило, допускают настройку своих панелей управления, делающую возможной работу в других разрешениях.

Цветовое разрешение (глубина цвета) определяет количество различных оттенков, которые может принять отдельная точка экрана. Максимально возможное цветовое разрешение зависит от свойств видеоадаптера и, в первую очередь, от количества имеющейся в нем видеопамати. Кроме того, оно зависит и от установленного разрешения экрана. При высоком разрешении экрана на каждую

точку изображения приходится отводить меньше места в видеопамяти, так что информация о цветах вынужденно оказывается более ограниченной.

Глубина цвета, соответствующая минимальным требованиям на сегодняшний день – 256 цветов, но большинство программ сейчас требуют не менее 65 тыс. цветов для эффективной работы (режим High Color). Наиболее комфортная работа достигается при глубине цвета 16,7 млн цветов (режим True Color). Работа в таком режиме (True Color) с высоким экранным разрешением требует использования значительных размеров видеопамяти. Современные видеоадаптеры способны не только хранить нужные данные, но также способны выполнять функции обработки изображения, снижая тем самым нагрузку на центральный процессор. Объем видеопамяти, установленной на видеоадаптер, сегодня определяется не только размером буфера кадра, но и необходимостью выполнения подобных дополнительных операций и обычно составляет от нескольких сотен мегабайт до нескольких гигабайт.

Видеоускорение – свойство видеоадаптера, которое заключается в том, что часть операций по построению изображений может происходить без выполнения математических вычислений в основном процессоре компьютера, а чисто аппаратным путем – преобразованием данных в микросхемах *видеоускорителя*. Видеоускорители входят в состав видеоадаптера (в таких случаях можно сказать, что видеокарта будет обладать функцией аппаратного ускорения). Различают два типа видеоускорителей – ускорители плоской (2D) и трехмерной (3D) графики. Ускорители первого типа наиболее эффективны для работы с прикладными программами, использующими стандартный интерфейс (обычно офисного применения), и оптимизированы для операционной системы Windows, а вторые ориентированы на работу с мультимедийными развлекательными программами, в первую очередь компьютерными играми, и профессиональными программами обработки трехмерной графики. Обычно в этих случаях используют разнообразные математические принципы для автоматизации выполнения графических операций. Все современные видеокарты обладают функциями как двумерного, так и трехмерного ускорения.

Звуковая карта – одно из устройств, созданное для улучшения пользовательских характеристик компьютера (рисунок 2.18).

Она устанавливается в разъем материнской платы в виде дочерней карты и предназначена для выполнения вычислительных операций, связанных с обработкой звука, речи, музыки.

К отдельному выходу звуковой карты подключаются внешние звуковые колонки через которые воспроизводится звук. Специальный разъем позволяет отправить звуковой сигнал на внешний усилитель. Имеется также разъем для подключения микрофона, что позволяет записывать речь или музыку и сохранять их на жестком диске для последующей обработки и использования.

Основным параметром звуковой карты – является *разрядность*. Она определяет количество битов, которые используются для преобразования сигнала из аналоговой формы в цифровую, и наоборот. Чем выше разрядность устрой-

ства, тем меньше погрешность, связанная с оцифровкой, и выше качество звучания. Минимальным показателем сегодня являются 16 разрядов, а наибольшее распространение имеют 32-разрядные и 64-разрядные устройства.

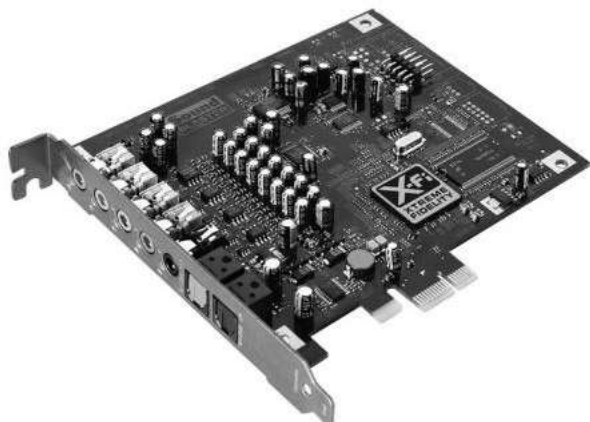


Рисунок 2.18 – Звуковая карта

В последние годы обработка звука считается относительно простой операцией, которую можно возложить на процессор, учитывая его возросшую мощность. Если пользователь не предъявляет повышенных требований к качеству звука, то можно использовать *интегрированные звуковые системы*, в которых функции обработки звука выполняются центральным процессором и микросхемами материнской платы. Колонки или иное устройство воспроизведения звука в этом случае подключается к гнездам, установленным непосредственно на материнской плате.

2.3.3 Системы, расположенные на материнской плате

Процессор – основная микросхема компьютера, которая предназначена для проведения вычислений (рисунок 2.19).

Конструктивно процессор состоит из ячеек, похожих на ячейки оперативной памяти. Данные в этих ячейках могут не только храниться, но и изменяться. Внутренние ячейки процессора называют *регистрами*. Данные, попавшие в некоторые особые регистры, рассматриваются не как данные, а как команды, управляющие обработкой данных в других регистрах. Среди регистров процессора есть и такие, которые в зависимости от своего содержания способны модифицировать исполнение команд. Таким образом, управляя распределением данных в разные регистры процессора, можно управлять обработкой данных. На этом и основано исполнение программ.

С остальными устройствами компьютера, и в первую очередь с оперативной памятью, процессор связан особыми группами проводников, называемых шинами. Основных шин три: *шина данных, адресная шина и командная шина*.



Рисунок 2.19 – Процессоры

Если два процессора имеют одинаковую систему команд, то они полностью совместимы на программном уровне. Это означает, что написанная для одного процессора программа, может быть исполнена и другим процессором. Процессоры, имеющие разные системы команд, как правило, несовместимы или ограниченно совместимы на программном уровне.

Группы процессоров, обладающие ограниченной совместимостью, рассматривают как *семейства процессоров*. Так, например, все процессоры Intel Pentium относятся к так называемому семейству x86. Родоначальником этого семейства был 16-разрядный процессор Intel 8086, на базе которого была собрана первая модель компьютера IBM PC. Впоследствии были разработаны процессоры Intel 80286, Intel 80386, Intel 80486, несколько моделей Intel Pentium; несколько моделей Intel Pentium MMX, модели Intel Pentium Pro, Intel Pentium II, Intel Celeron, Intel Xeon, Intel Pentium III, Intel Pentium 4, Intel Core и другие. Все перечисленные выше модели, а также многие модели процессоров компании AMD и некоторых других производителей относятся к семейству x86 и обладают совместимостью по принципу «сверху вниз».

Принцип совместимости «сверху вниз» – это пример неполной совместимости, который основан на том, что новый процессор «понимает» все команды своих предшественников, но старые модели не поддерживают команды новых процессоров.

Основными параметрами процессоров являются: *рабочее напряжение, разрядность, рабочая тактовая частота, коэффициент внутреннего умножения тактовой частоты и размер кэш-памяти*.

Оперативная память (RAM – Random Access Memory) – это массив кристаллических ячеек, способных хранить данные (рисунок 2.20).

Существует много различных типов оперативной памяти, но с точки зрения физического принципа действия различают *динамическую память* (DRAM) и *статическую память* (SRAM).

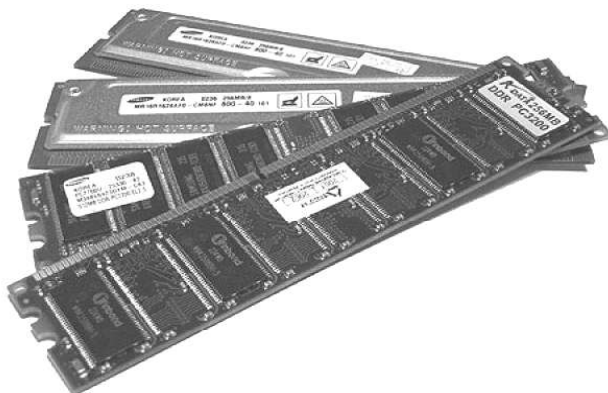


Рисунок 2.20 – Оперативная память

Ячейки динамической памяти (DRAM) можно представить в виде микроконденсаторов, способных накапливать заряд на своих обкладках. Это наиболее распространенный и экономически доступный тип памяти. Недостатки этого типа связаны, во-первых, с тем, что как при заряде, так и при разряде конденсаторов неизбежны переходные процессы, то есть запись данных происходит сравнительно медленно. Второй важный недостаток связан с тем, что заряды ячеек имеют свойство рассеиваться в пространстве, причем весьма быстро. Если оперативную память постоянно не «подзаряжать», утрата данных происходит через несколько сотых долей секунды. Для борьбы с этим явлением в компьютере происходит постоянная *регенерация (освежение, подзарядка)* ячеек оперативной памяти. Регенерация осуществляется несколько десятков раз в секунду и вызывает непроизводительный расход ресурсов вычислительной системы.

Ячейки статической памяти (SRAM) можно представить, как электронные микроэлементы – *триггеры*, состоящие из нескольких транзисторов. В триггере хранится не заряд, а состояние (*включен/выключен*), поэтому этот тип памяти обеспечивает более высокое быстродействие, хотя технологически он сложнее и, соответственно, дороже.

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

Каждая ячейка памяти имеет свой адрес, который выражается числом. В большинстве современных процессоров предельный размер адреса обычно составляет 32 разряда, а это означает, что всего независимых адресов может быть 2^{32} . Одна адресуемая ячейка содержит восемь двоичных ячеек, в которых можно сохранить 8 бит, то есть один байт данных.

Таким образом, в современных компьютерах возможна *непосредственная*

адресация к полю памяти размером 2^{32} байт = 4 Гбайт. Однако это отнюдь не означает, что именно столько оперативной памяти непременно должно быть в компьютере. Предельный размер поля оперативной памяти, установленной в компьютере, определяется микропроцессорным комплектом (*чипсетом*) материнской платы и обычно не может превосходить нескольких гигабайт. Минимальный объем памяти определяется требованиями операционной системы и для современных компьютеров составляет 1 Гбайт.

Представление о том, сколько оперативной памяти должно быть в типовом компьютере, непрерывно меняется. В середине 80-х годов поле памяти размером 1 Мбайт казалось огромным, в начале 90-х годов достаточным считался объем 4 Мбайт, к середине 90-х годов он увеличился до 8 Мбайт, а затем и до 16 Мбайт. Сегодня типичным считается размер оперативной памяти в 1 Гбайт, но тенденция к росту сохраняется.

Оперативная память в компьютере размещается на стандартных панельках, называемых *модулями*. Модули оперативной памяти вставляют в соответствующие разъемы на материнской плате. Если к разъемам есть удобный доступ, то операцию можно выполнять своими руками. Если удобного доступа нет, может потребоваться неполная разборка узлов системного блока, и в таких случаях операцию поручают специалистам.

Основными характеристиками модулей оперативной памяти являются объем памяти и скорость передачи данных. Скорость передачи данных определяет максимальную пропускную способность памяти (в мегабайтах в секунду или гигабайтах в секунду) в оптимальном режиме доступа. При этом учитывается время доступа к памяти, ширина шины и дополнительные возможности, такие как передача нескольких сигналов за один такт работы. Одинаковые по объему модули могут иметь разные скоростные характеристики.

Иногда в качестве определяющей характеристики памяти используют *время доступа*. Оно измеряется в миллиардных долях секунды (наносекундах, нс). Для современных модулей памяти это значение может составлять 5 нс, а для особо быстрой памяти, используемой в основном в видеокартах, – снижаться до 2 – 3 нс.

2.4 Контрольные вопросы

- 1 Охарактеризуйте поколения компьютеров: нулевое, первое, второе, ...
- 2 Опишите классификации персональных компьютеров.
- 3 Охарактеризуйте базовую конфигурацию компьютера.
- 4 Опишите технические характеристики монитора.
- 5 Клавиатура. Основные группы клавиш.
- 6 Какие устройства находятся в системном блоке.
- 7 Какие платы используются в персональных компьютерах.
- 8 Опишите функции и характеристики процессора.
- 9 Опишите принципы работы оперативной памяти компьютера.

Список источников

- 1 Грошев А. С. Информатика : учебник для вузов. – Архангельск : Изд-во Арханг. гос. техн. ун-та, 2010. – 470 с.
- 2 Макарова Н. В., Волков В. Б. Информатика : учебник для вузов. – Санкт-Петербург : Питер, 2011. – 576 с.
- 3 Симонович С. В. Информатика. Базовый курс : учебник для вузов. – 3-е изд. – Санкт-Петербург : Питер, 2011. – 640 с.
- 4 Степанов А. Н. Информатика : учебник для вузов. – 6-е изд. – Санкт-Петербург : Питер, 2010. – 720 с.
- 5 Таганов Л. С. Информатика : учеб. пособие. – Кемерово : Изд-во Кузбас. гос. техн. ун-та, 2010. – 330 с.

3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ

Возможности конкретного компьютера по обработке данных и область его применения определяются используемым на нем программным обеспечением.

Программное обеспечение (ПО) – совокупность программ, установленных на компьютере.

3.1 КЛАССИФИКАЦИЯ ПО ЭВМ

Программное обеспечение ЭВМ можно классифицировать по различным признакам.

В зависимости от области применения программных продуктов выделяют три типа программного обеспечения: системное, прикладное и инструментальное ПО.

Каждый тип программного обеспечения выполняет различные функции и взаимодействует с другими типами ПО (рисунок 3.1):

- *системное ПО* обеспечивает и контролирует доступ к аппаратному обеспечению компьютера;
- *прикладное ПО* взаимодействует с аппаратным обеспечением через системное ПО и обеспечивает решение задач пользователя в различных предметных областях;
- *инструментальное ПО* также работает с аппаратным обеспечением через системное ПО и позволяет создавать новые программы для компьютера.



Рисунок 3.1 – Структура программного обеспечения компьютера

3.2 СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Системное программное обеспечение управляет всеми ресурсами ЭВМ и осуществляет общую организацию процесса обработки информации. Оснащение компьютера системным ПО подготавливает его к установке прикладных программ, а также к взаимодействию всех программ с оборудованием и пользователем.

Основу системного ПО составляют *операционные системы*, управляющие всеми аппаратными узлами компьютера и обеспечивающие взаимодействие между аппаратным обеспечением компьютера, его программным обеспечением и пользователем.

В состав системного ПО входит большое количество *утилит* - программ вспомогательного назначения, расширяющих возможности операционной системы.

К системным программам относятся *драйверы* – специальные программы, отвечающие за взаимодействие с конкретными устройствами. При подключении к ЭВМ нового оборудования должен быть установлен драйвер, обеспечивающий для других программ взаимосвязь с этим оборудованием. Драйверы обычно входят в состав операционной системы.

В состав системных программ входят *операционные оболочки* (или файловые менеджеры). Они обеспечивают альтернативный способ общения с компьютером, отличный от штатных средств операционной системы.

Средства контроля и диагностики, также входящие в состав системного ПО, обеспечивают автоматическую проверку функционирования отдельных узлов ЭВМ.

3.2.1 Операционная система: понятие, основные функции, структура

Операционная система (ОС) – это комплекс программных средств, обеспечивающих выполнение других программ, распределение ресурсов, ввод-вывод и управление данными. Операционная система была создана для того, чтобы автоматизировать работу пользователя и скрыть от него сложности общения с аппаратурой.

Операционная система выполняет две главные группы функций:

1 Поддержка работы пользователя:

- определяет интерфейс пользователя (среду взаимодействия пользователя с компьютером);
- обеспечивает разделение аппаратных ресурсов между пользователем и общими данными и дает возможность работы с ними в режиме коллективного пользования.

2 Поддержка работы всех программ и организация их взаимодействия с устройствами компьютера:

- обеспечение эффективного выполнения операций ввода-вывода информации (связь с устройствами ввода-вывода);
- организация хранения данных;
- распределение ресурсов (оперативной памяти, процессора, внешних устройств);
- обеспечение взаимодействия программ и данных, а также взаимодействие программ друг с другом;
- управление работой устройств ЭВМ;
- выявление различных событий, возникающих в ходе работы, и соответствующая реакция на них.

Кроме базовых функций операционные системы могут предоставлять различные дополнительные функции:

- поддерживать работу локальной компьютерной сети без специального программного обеспечения;
- обеспечивать доступ к основным службам Интернета встроенными в ОС средствами;

- предоставлять средства защиты данных от несанкционированного доступа;
- обеспечение работы нескольких пользователей на одном компьютере с сохранением персональных настроек рабочей среды каждого из них;
- обеспечивать работу с компьютером лицам, имеющим физические недостатки.

Все перечисленные функции реализуются наборами программ в составе операционной системы, образующими отдельные модули или подсистемы. Наиболее важными подсистемами ОС являются:

1 Подсистема управления процессами. Для каждой выполняемой программы ОС организует один или более процессов (задач). Подсистема управления процессами планирует очередность выполнения процессов, обеспечивает их необходимыми ресурсами, обеспечивает взаимодействие и синхронизацию процессов.

2 Подсистема управления памятью распределяет физическую память между всеми существующими в системе процессами.

3 Подсистема управления файлами предоставляет пользователям удобные средства для работы с файлами. Функции управления файлами сосредоточены в файловой системе ОС.

4 Подсистема управления внешними устройствами (подсистема ввода-вывода) обеспечивает взаимодействие между ядром компьютера и всеми подключенными к нему устройствами с помощью драйверов – программ, управляющих устройствами.

5 Подсистема защиты данных и администрирования обеспечивает защиту от сбоев и отказов аппаратуры и ошибок программного обеспечения, а также предоставляет средства защиты от несанкционированного доступа.

6 Интерфейс прикладного программирования упрощает разработку новых приложений для конкретной ОС.

7 Пользовательский интерфейс. Современные ОС поддерживают два вида пользовательского интерфейса:

- неграфический – реализует интерфейс командной строки. Работа пользователя основана на вводе команд управления с клавиатуры;
- графический – в качестве устройства управления кроме клавиатуры используется мышь. Работа пользователя основана на взаимодействии активных и пассивных экранных элементов управления (окон, иконок, кнопок и т. п.).

3.2.2 Архитектура ОС

Под *архитектурой ОС* понимают состав, назначение и порядок взаимодействия модулей, реализующих основные функции ОС.

Какой-либо единой архитектуры ОС не существует, но существуют универсальные подходы к структурированию ОС.

Наиболее общим подходом к структуризации является разделение всех модулей ОС на две группы [3]:

- ядро – модули, выполняющие основные функции ОС;
- модули, выполняющие вспомогательные функции ОС.

Ядро ОС выполняет такие базовые функции ОС, как управление процессами, памятью, устройствами ввода-вывода и т. п. В ядре решаются внутри-системные задачи организации вычислительного процесса, недоступные для приложений.

Приложения взаимодействуют с аппаратурой только через ядро ОС. Приложения могут обращаться к ядру с запросами – системными вызовами – для выполнения тех или иных действий, например, открытие и чтение файла, получение системного времени, вывода информации на дисплей и т. д. Функции ядра, которые могут вызываться приложениями, образуют прикладной программный интерфейс – API (Application Programming Interface).

Ядро может состоять из следующих слоев:

1 Средства аппаратной поддержки ОС – это аппаратные устройства компьютера, которые непосредственно участвуют в организации вычислительных процессов. К ним относятся: система прерываний, средства поддержки привилегированного режима, средства поддержки виртуальной памяти, средства защиты памяти и др.;

2 Машинно-зависимые компоненты ОС – это программные модули, в которых отражается специфика аппаратной платформы компьютера. Эти модули экранируют вышележащие слои ОС от особенностей аппаратуры, что позволяет разрабатывать эти слои в единственном экземпляре для всех типов аппаратных платформ, поддерживаемых данной ОС;

3 Базовые механизмы ядра – выполняют наиболее примитивные операции ядра, такие как диспетчеризацию прерываний, перемещение страниц из памяти на диск и обратно и т. п. Модули этого слоя не принимают решений о распределении ресурсов, а только обрабатывают решения, принятые модулями вышележащих уровней;

4 Менеджеры ресурсов – программные модули, управляющие ресурсами вычислительной системы. Это менеджеры (диспетчеры) процессов ввода-вывода, оперативной памяти и файловой системы. Каждый менеджер ведет учет свободных и используемых ресурсов и планирует их распределение в соответствии запросами приложений;

5 Интерфейс системных вызовов – является самым верхним слоем ядра и взаимодействует непосредственно с приложениями и системными утилитами, образуя прикладной программный интерфейс ОС (API). Функции API, обслуживающие системные вызовы, предоставляют доступ к ресурсам системы в удобной компактной форме, без указания деталей их физического расположения.

Для обеспечения высокой скорости работы ОС модули ядра или большая их часть постоянно находятся в оперативной памяти компьютера. Кроме того ядро ОС работает в привилегированном режиме, обеспечивающем безопасность работы ОС от вмешательства приложений.

Вспомогательные модули обычно подразделяются на группы:

- утилиты – программы, выполняющие отдельные задачи управления и сопровождения вычислительной системы;
- системные обрабатывающие программы – текстовые и графические редакторы (Paint, Notepad в ОС Windows), компиляторы и др.;
- программы предоставления пользователю дополнительных услуг (калькулятор, игры, средства мультимедиа);
- библиотеки процедур, упрощающие разработку приложений (например, библиотека функций ввода-вывода, библиотека математических функций и т. п.).

Вспомогательные модули ОС оформляются как обычные приложения, они обращаются к функциям ядра посредством системных вызовов и выполняются в пользовательском режиме. В этом режиме запрещается выполнение некоторых команд, которые связаны с функциями ядра ОС (управление ресурсами, распределение и защита памяти и т. п.).

Разделение операционной системы на ядро и модули-приложения обеспечивает легкую расширяемость ОС. Чтобы добавить новую высокоуровневую функцию, достаточно разработать новое приложение, при этом не требуется модифицировать функции, образующие ядро системы.

3.2.3 Классификация ОС и краткий обзор

Операционные системы могут различаться особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера, типами аппаратных платформ, областями использования и многими другими свойствами.

Ниже приведена классификация ОС по основным признакам:

1 По назначению ОС делятся на универсальные и специализированные. Специализированные ОС, как правило, работают с фиксированным набором программ и эксплуатируются вычислительными системами, на которых невозможно использовать универсальные ОС по соображениям эффективности, надежности, защищенности и т. п., а также вследствие специфики решаемых задач. Универсальные ОС рассчитаны на решение любых задач пользователей.

2 По способу загрузки можно выделить загружаемые ОС (большинство) и системы, постоянно находящиеся в памяти вычислительной системы. Последние, как правило, специализированные и используются для управления работой специализированных устройств (например, спутников, научных приборов и др.).

3 По особенностям алгоритмов управления ресурсами. Так как главным ресурсом системы является процессор, то традиционно ОС классифицируют по алгоритмам управления процессором:

по числу одновременно выполняемых задач (программ) ОС делятся на однозадачные (MS-DOS) и многозадачные (OS/2, UNIX, Windows). Однозадачные ОС обеспечивают простое и удобное взаимодействие пользователя с компьютером. Они имеют средства управления файлами, периферийными устройствами и средства общения с пользователем. Многозадачные ОС, кроме того, управляют разделением совместно используемых ресурсов (процессор, память, файлы и т. д.);

- по числу одновременно работающих пользователей ОС делятся на однопользовательские (MS-DOS, Windows 3x) и многопользовательские (UNIX, Windows NT/XP/2000 и выше). Главное отличие многопользовательских систем от однопользовательских – наличие средств защиты информации каждого пользователя от несанкционированного доступа других пользователей;

- в зависимости от отсутствия или наличия средств поддержки *многопроцессорной обработки* выделяют ОС без поддержки мультипроцессорирования (Windows 3.x, Windows 95) и с поддержкой мультипроцессорирования (OS/2, UNIX, Windows NT/XP /2000 и выше). Многопроцессорные ОС классифицируются по способу организации вычислительного процесса на асимметричные ОС (выполняются на одном процессоре, распределяя прикладные задачи по остальным процессорам) и симметричные ОС (использует все процессоры, разделяя их между системными и прикладными задачами).

4 По области использования выделяют три типа ОС:

- *системы пакетной обработки* (OS/360, ОС ЕС) предназначались для решения задач в основном вычислительного характера, не требующих быстрого получения результатов, обеспечивая максимальную пропускную способность при хорошей загрузке всех ресурсов компьютера;

- *системы разделения времени* (UNIX, VMS) обеспечивают удобство и эффективность работы пользователя, который имеет отдельный терминал и может вести диалог со своей программой;

- *системы реального времени* (QNX, RT/11) предназначены для управления техническими объектами (станок, спутник, и т. п.), где существует предельное время на выполнение программ, управляющих объектом.

5 По аппаратной платформе (типу вычислительной техники), для которой они предназначаются, ОС делят на следующие группы:

- *операционные системы для смарт-карт*. Некоторые из них могут управлять только одной операцией, например, электронным платежом;

- *встроенные операционные системы* управляют карманными компьютерами, мобильными телефонами, телевизорами, микроволновыми печами и т. п.;

- *операционные системы для персональных компьютеров*, например, Windows XP, Windows 2010, Linux, Mac OSX и др.;

- *операционные системы мини-ЭВМ*, например, UNIX для PDP-7;

- *операционные системы мэйнфреймов* (больших машин), например, OS/390;

- *серверные операционные системы*, например, UNIX, Windows 2000, Linux. Область применения – локальные вычислительные сети, региональные сети, Internet.

3.2.4 Файловая система: основные понятия, функции

Вся информация в компьютере хранится в виде файлов, с которыми работает пользователь и операционная система.

Файл – это именованная область внешней памяти, в которую можно за-

писывать и из которой можно считывать данные [3].

Файл является учетной единицей информации в операционной системе. Любые действия с информацией в компьютере осуществляются над файлами: запись на диск, вывод на экран, ввод с клавиатуры, печать и т. д.

Все типы файлов имеют имена. Во многих операционных системах имя файла состоит из двух частей, разделенных точкой. Часть имени до последней точки представляет собой описательное название содержимого файла. Часть имени после последней точки называется расширением файла и обычно означает его тип (т. е. вид информации, содержащейся в нем).

Ограничения на используемые символы и их количество в имени файла определяются используемой операционной системой.

Примеры имен файлов:

- типовой договор.doc – текстовый документ;
- image.bmp – файл, содержащий изображение;
- game.exe – исполняемый файл.

В некоторых ОС, например, Windows, расширение указывает на программу, создавшую файл. Так первый файл в примере выше мог быть создан с помощью программы Microsoft Word. Когда пользователь открывает файл с зарегистрированным расширением, операционной системой автоматически запускается соответствующая этому расширению программа.

Понятие файла включает не только хранимые им данные и имя, но и информацию, описывающую свойства файла. Эта информация составляет *атрибуты* файла. Список атрибутов может быть различным в различных ОС.

Для характеристики файла могут использоваться следующие атрибуты:

- 1) объем файла в байтах;
- 2) дата и время создания файла;
- 3) дата и время модификации (или открытия) файла;
- 4) специальные атрибуты файла:

- «только для чтения» – наличие такого атрибута защищает файл от изменений; для изменения или удаления такого файла требуется предварительно снять данный атрибут;

- «скрытый» – файлы с этим атрибутом по умолчанию не отображаются на экране при проведении файловых операций;

- «системный» – файлы с этим атрибутом обладают важными функциями в работе самой операционной системы;

- «архивируемый» – устанавливается при создании файла и сбрасывается программами резервного копирования для обозначения того, что копия файла помещена в архив. Поэтому наличие атрибута «архивируемый» обычно значит, что для файла не было сделано резервной копии.

Файлы создаются для долговременного хранения информации на носителях информации. Основным носителем информации в компьютере является жесткий диск.

Каждому диску присваивается имя, которое задается латинской буквой с

двоеточием, например, А:, В:, С:, D: и т. д. Принято, что А: и В: – это накопители на гибких магнитных дисках, а С: и D: и т. д. – накопители на жестких магнитных дисках, накопители на оптических дисках или электронные диски. Диск, на котором записана операционная система, называется системным диском и обычно имеет имя С:.

Имена файлов регистрируются на дисках в каталогах, также называемых директориями или папками.

Каталог – это специальный файл, который хранит сведения о других файлах и каталогах, логически находящихся в данном каталоге (имя, размер, дату создания, адрес файла на диске и т. д.).

Совместно с термином «каталог» часто используются термины «корневой», «текущий», «родительский», «подкаталог».

Корневой каталог – это самый главный каталог диска, фактически это и есть диск, т. к. корневой каталог содержит в себе абсолютно все файлы, созданные на диске.

Текущий каталог – это каталог, с которым в настоящий момент работает пользователь. Как только пользователь откроет какой-либо другой каталог, он тут же становится текущим.

Родительский каталог – каталог, в котором содержится текущий каталог.

Подкаталог – каталог, находящийся в другом каталоге. Любой каталог может содержать произвольное число подкаталогов, в каждом из которых могут храниться файлы и другие каталоги. Благодаря развитой системе подкаталогов создается иерархическая структура файлов на диске.

Когда используется файл не из текущего каталога, необходимо указать, в каком каталоге этот файл находится. Это можно сделать с помощью указания пути к файлу.

Путь к файлу – это цепочка каталогов, которую необходимо пройти по иерархической структуре к каталогу, где зарегистрирован искомый файл. При задании пути имена каталогов записываются в порядке следования и отделяются друг от друга символами « \ ». Например, строка С:\Temp\Text задает путь к файлам, находящимся в подкаталоге Text каталога Temp диска С:.

Полное имя файла – это сочетание пути к файлу и его имени. Например, строка Z:\Documents\User\Programs\readme.txt однозначно определяет файл readme.txt в иерархической структуре диска Z:.

Удобные для пользователя средства работы с файлами предоставляются в ОС файловой системой.

Файловая система (ФС) – это часть операционной системы, включающая [3]:

- совокупность всех файлов на носителе информации;
- наборы структур данных, используемых для управления файлами (каталоги, таблицы распределения свободного и занятого пространства на диске и др.);
- комплекс системных программных средств, реализующих различные операции над файлами.

Главная задача ФС – организация эффективной работы с данными, хранящимися на диске, и обеспечение удобного интерфейса пользователя при работе с этими данными.

Основными функциями файловой системы в многопользовательских многозадачных ОС являются:

- создание файлов и каталогов (папок) и их именование;
- копирование и перемещение файлов между дисками и между каталогами (папками) одного диска;
- удаление файлов и каталогов (папок);
- навигация по файловой структуре с целью доступа к заданному файлу, каталогу (папке);
- минимизация или полное исключение возможных потерь или повреждений данных;
- поддержка ввода-вывода для различных типов устройств хранения информации;
- обеспечение поддержки совместного использования файлов несколькими приложениями и/или пользователями;
- защита файлов от несанкционированного доступа;
- управление атрибутами файлов.

Задачи, решаемые файловой системой, во многом определяются способом организации вычислительного процесса (наиболее простые – в однопрограммных и однопользовательских ОС, наиболее сложные – в сетевых ОС).

3.2.5 Сервисные программы

Сервисные программы предназначены для расширения или улучшения функций операционных систем, для обеспечения более комфортной работы пользователя. К сервисным программам относят операционные оболочки (или файловые менеджеры, программы-оболочки) и широкий класс программ вспомогательного назначения, называемых утилитами (лат. *utilitas* – польза).

Первые операционные оболочки были созданы для преобразования командного пользовательского интерфейса в графический интерфейс. Вместо запоминания и ввода управляющих команд с клавиатуры они предоставляют пользователю удобный и наглядный доступ к файлам и каталогам и различные сервисные услуги, используя окна и меню команд.

Для организации работы с файлами операционные оболочки имеют двухпанельный однооконный интерфейс с командной строкой и большим набором «горячих» клавиш, существенно ускоряющих выполнение многих операций. Благодаря большому количеству дополнительных функций (удобная работа с архивами, с документами и т. д.) позволяют отказаться от использования некоторых утилит.

Основные функции операционных оболочек:

- предоставление интуитивно понятного, удобного, наглядного интерфейса;

- выполнение основных операций с файлами (копирование, перенос, удаление, создание, поиск и др.);
- вывод информации о выбранном пользователем файле или каталоге;
- поддержка встроенных архиваторов;
- поддержка сетевых протоколов, включая Internet-протоколы;
- поддержка встроенных средств обслуживания дисков, дублирующих стандартные средства ОС.

К самым популярным программам данного класса относятся:

1 Norton Commander – самый первый файловый менеджер от Питера Нортона;

2 Far Manager – псевдографическое приложение ОС Windows, очень похожее на Norton Commander;

3 Total Commander (панель Windows Commander) – создан по образу Norton Commander.

Важным классом системных программ являются также сервисные программы вспомогательного назначения – утилиты. Они либо расширяют и дополняют соответствующие возможности операционной системы, либо решают самостоятельные важные задачи.

Основные типы утилит:

1 Антивирусные программы – предназначены для предотвращения заражения компьютерными вирусами и ликвидации последствий заражения вирусами. Антивирусные программы можно разделить на несколько видов:

- программы-детекторы осуществляют поиск характерной для конкретного вируса сигнатуры в оперативной памяти и в файлах и при обнаружении выдают соответствующее сообщение;

- программы-доктора находят зараженные вирусами файлы и удаляют из файла тело программы-вируса, возвращая файл в исходное состояние;

- программы-ревизоры запоминают исходное состояние программ, каталогов и системных областей диска, а затем периодически или по желанию пользователя сравнивают текущее состояние с исходным, сообщая об обнаруженных изменениях;

- программы-фильтры – это резидентные программы, предназначенные для обнаружения при работе компьютера подозрительных действий, характерных для вирусов. При попытке какой-либо программы произвести подозрительные действия выводят сообщение с предложением запретить или разрешить соответствующее действие;

- вакцины – это резидентные программы, предотвращающие заражение файлов. Вакцина модифицирует программу или диск таким образом, чтобы это не отражалось на их работе, а вирус будет воспринимать их зараженными и поэтому не внедрится.

Современные антивирусы представляют собой многофункциональные продукты, сочетающие в себе как профилактические средства, так и средства лечения вирусов и восстановления данных.

2 Программы-упаковщики (архиваторы) – позволяют за счет специальных методов сжатия создавать копии файлов меньшего размера и объединять копии нескольких файлов в один архивный файл, а также распаковывать архивы (извлекать файлы из архива). Архиваторы часто используют для создания резервных копий данных. Наиболее популярные форматы архивов:

- ZIP – отличается приемлемой степенью сжатия информации и достаточно высоким быстродействием. Сегодня он фактически является стандартом для передачи данных в Internet, и его в обязательном порядке поддерживают практически все программы-архиваторы;

- RAR – позволяет получить размер сжатого файла гораздо меньший, чем ZIP, но требует более продолжительного времени для обработки архива;

- CAB – применяется в продуктах Microsoft как стандартный для упаковки файлов, имеет высокий коэффициент сжатия;

- GZIP, TAR – наиболее распространены в ОС Unix и Linux.

3 Программы для обслуживания дисков, позволяющие решить следующие задачи:

- форматирование дисков, при этом все содержимое дисков уничтожается;

- проверка диска на наличие логических и физических ошибок, которые могут возникнуть из-за сбоев в работе программ и оборудования, некорректного завершения работы компьютера;

- дефрагментация диска – операция, при выполнении которой файлы, фрагментами расположенные в разных частях диска, собираются в единое целое, что приводит к сокращению времени их чтения и записи;

- очистка диска – удаление пользовательского и системного «мусора» для освобождения дискового пространства.

4 Программы контроля, тестирования и диагностики устройств компьютера используются для проверки правильности функционирования устройств компьютера и для обнаружения неисправностей в процессе эксплуатации, по возможности указывают причину и место неисправности.

5 Программы восстановления информации.

6 Коммуникационные программы – предназначены для установления соединений с удалёнными компьютерами, для обслуживания передачи сообщений электронной почты, обеспечения пересылки факсимильных сообщений и множества других операций в компьютерных сетях.

7 Программы для записи оптических дисков и многие другие.

Часть утилит входит в состав операционной системы (как правило, это программы обслуживания), другая часть функционирует независимо от нее.

3.3 ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Прикладное ПО предназначено для решения пользовательских задач из различных предметных областей.

Единой классификации прикладного ПО не существует в связи с огромным многообразием программных средств этого класса. Чаще всего выделяют следующие группы прикладного ПО:

- прикладное ПО общего назначения;
- проблемно-ориентированное прикладное ПО;
- интегрированные системы.

3.3.1 Прикладное ПО общего назначения

Прикладное ПО общего назначения ориентировано на широкий круг задач из различных предметных областей, позволяет автоматизировать наиболее часто используемые работы пользователя.

К данной группе ПО относятся:

1 Текстовые редакторы – это наиболее широко используемый вид прикладных программ. Основные функции – это ввод и редактирование текстовых данных. Все многообразие текстовых редакторов условно можно разделить на три основных вида:

- простейшие текстовые редакторы, обладающие минимумом возможностей и способные работать с документами без форматирования (оформления) в обычном текстовом формате TXT. Примеры таких редакторов: NotePad (Блокнот), KEdit, KWrite, EditPad и другие;

- текстовые редакторы, обеспечивающие ограниченный набор возможностей по оформлению документов: использование шрифтов, абзацев, списков, вставка изображений. Они работают со всеми стандартными текстовыми файлами (TXT, RTF, DOC). К таким программам можно отнести входящий в ОС Windows редактор WordPad, Microsoft Works, Лексикон;

- текстовые процессоры – выполняют практически все операции с текстом. Они позволяют использовать различные шрифты, абзацы любой формы, автоматический перенос слов, вставку рисунков и таблиц, автоматическую нумерацию страниц, оформление текста в несколько столбцов, строить оглавления, проверять правописание и другие операции. Примеры: Microsoft Word, OpenOffice.org Writer, Adobe In Copy, WordExpres.

2 Графические редакторы позволяют создавать и редактировать рисунки. В простейших редакторах есть возможность рисования линий, кривых, раскраски областей экрана, создания надписей. В редакторах, ориентированных на обработку фотоизображений, имеются средства для настройки яркости и контрастности изображения, ретуширования изображения и др. Различают три вида графических редакторов:

- растровые графические редакторы – в них изображение представляется в виде прямоугольной матрицы, каждая ячейка которой - цветная точка (пиксель). Простейшие растровые графические редакторы позволяют рисовать растровые изображения (Paint, Paint.net, PaintBrush). Другой вид растровых графических редакторов предназначен для обработки готовых растровых изображений, полученных в результате сканирования фотографий, в результате съемки с помощью

цифрового фотоаппарата (Adobe Photoshop, Corel PhotoPaint, Gimp);

- векторные редакторы работают с изображением, состоящим из различных геометрических объектов – линий, окружностей, прямоугольников и т. д. Форма, цвет и пространственное положение составляющих изображение объектов описывается с помощью математических формул (например, для прямой линии задаются координаты двух точек, для круга – координаты центра и радиус). Примеры векторных редакторов: Adobe Illustrator, CorelDRAW, Macromedia FreeHand, Xara Xtreme, OpenOffice.org Draw;

- 3-D редакторы (программы трехмерной графики) – позволяют создавать трехмерные объекты любой формы, подвергать их деформациям изгиба, кручения и сдвига. Используются для изготовления спецэффектов для кино и телевидения, получения реалистичных фотоизображений. Примеры программ: Alias Maya, 3D Studio MAX, Ulead Cool3D, Production Studio, Adobe Acrobat 3D.

3 Табличные процессоры (электронные таблицы) предназначены для хранения и обработки данных, представленных в табличном виде. При работе с табличным процессором на экран выводится прямоугольная таблица, в клетках которой могут находиться числа, пояснительные тексты и формулы для расчета. Табличные процессоры позволяют пересчитывать значения элементов таблицы по заданным формулам, строить разнообразные графики. С помощью данных программ можно решать экономические, математические и статистические задачи. Примеры электронных таблиц: Microsoft Excel, SuperCalc, OpenOffice.org Calc, Quattro Pro, Lotus 1-2-3 for Windows.

4 Системы управления базами данных (СУБД). Базой данных называют большие массивы логически связанных данных. СУБД предназначены для создания структуры базы данных, предоставления средств для заполнения этой структуры, обеспечения возможности доступа к данным, а также предоставления средств поиска и фильтрации данных. Примеры СУБД: MS Access, dBase, Paradox, MySQL, Oracle.

5 Программы подготовки презентаций предназначены для создания компьютерных презентаций, представляющих собой набор слайдов, последовательность показа которых может меняться в процессе демонстрации презентации. Каждый слайд может включать в себя текст, таблицы, диаграммы, изображения, звук, анимацию и др. Презентации обычно используют в процессе выступлений на конференциях, при объяснении нового материала преподавателем и т. д. Примерами таких программ являются Microsoft PowerPoint, OpenOffice.org Impress, StarOffice Impress, Autodesk Animator Pro и другие.

Приведенная классификация является условной и неполной, т. к. постоянно появляются новые прикладные программы или совершенствуются уже существующие.

3.3.2 Проблемно-ориентированное прикладное ПО

Проблемно-ориентированное прикладное ПО используется для решения

определенных классов задач в конкретных предметных областях (научных, инженерных, издательских, экономических и многих других).

Ниже перечислены некоторые виды проблемно-ориентированного ПО:

1 *Бухгалтерские программы* – предназначены для ведения бухучета, подготовки финансовой отчетности, финансового анализа деятельности предприятий. Примеры российских бухгалтерских программ: 1С, Парус, Галактика, Инфо-бухгалтер.

2 *Издательские системы* – автоматизируют процесс верстки полиграфических изданий. Предназначены для оформления журналов, газет, подготовки рекламных буклетов и пр. В отличие от текстовых редакторов издательские системы в первую очередь предназначены для оформления документов, а не для ввода и редактирования. Основная их функция - верстка, т. е. размещение текста и изображений по страницам документа. Примеры издательских систем: Microsoft Publisher, QuarkXPress, Macromedia FreeHand, Adobe PageMaker, Adobe Illustrator, Adobe InDesign и другие.

3 *Системы автоматизированного проектирования (САПР)* – предназначены для решения задач проектирования изделий, устройств или архитектурных сооружений. Имеют средства создания, редактирования и печати чертежей, позволяют выполнять инженерные расчеты. Примеры: AutoCAD, Palasm (проектирование матриц ПЗУ), AUTRA DSP (комплексная: машиностроение, строительство и др.).

4 *ПО решения задач прикладной математики и статистики* – реализует методы прикладной математики (MathCAD, MatLab, Matematica, Maple V и др.) и математической статистики (Statistica, StatGraf, StatWare, DataStat, MuliStat и др.). Некоторые математические пакеты позволяют выполнять не только численное решение задач, но и символьные операции над формулами (алгебраические преобразования, дифференцирование, интегрирование).

5 *Экспертные системы* – предназначены для анализа данных, содержащихся в базах знаний и выдачи результатов по запросу пользователя. Такие системы используются, когда для принятия решения нужны широкие специальные знания. Применяются в химии, медицине, фармакологии, юриспруденции. Экспертные системы: MYCIN (медицинская), TOPSI (принятия решений), Expert Choice (оптимизации решений).

6 *Системы автоматизированного перевода* – позволяют переводить отдельные слова и строить смысловые связи в предложениях, не всегда учитывая те или иные особенности языка, поэтому они предназначены лишь для общего ознакомления с содержанием документа. Примеры: Prompt XT, Magic Gooddy и другие.

7 *Программы обработки аудиоданных* – позволяют прослушивать, создавать, редактировать и выполнять преобразование форматов аудиоданных. Примеры: Cool Edit, Sound Forge, Visual Arranger for Windows.

Системы видеомонтажа – предназначены для цифровой обработки видеоматериалов, монтажа, создания видеоэффектов, добавления звука, титров и т. д.

Примеры: Adobe Premiere, Pinnacle Studio.

8 *Браузеры* – средства просмотра электронных документов, получаемых с веб-сайтов. Воспроизводят текст, графику, музыку, человеческий язык, позволяют работать с электронной почтой. Примеры браузеров: Microsoft Internet Explorer, Mozilla Firefox, Opera и другие.

3.3.3 Интегрированные системы

Интегрированные системы объединяют основные функции программ общего назначения, используя единый информационный интерфейс. В таких системах используются единые методы доступа к файлам отдельных программ, единые методы выполнения базовых операций по обработке данных. Примером интегрированных систем являются Microsoft Office, LibreOffice, OpenOffice.

Наиболее популярной интегрированной системой является Microsoft Office. Она представляет собой семейство прикладных программных продуктов, которое объединяет различные приложения в универсальную среду для работы с информацией:

1 Microsoft Word – текстовый процессор, позволяющий создавать профессионально оформленные документы, содержащие рисунки, схемы, формулы, таблицы, диаграммы.

2 Microsoft Excel – программа для работы с электронными таблицами, располагающая разнообразными средствами для форматирования, отображения, преобразования и анализа данных, проведения математических, финансовых, статистических и других вычислений.

3 Microsoft PowerPoint – программа, предназначенная для создания презентаций, представляющих собой совокупность слайдов, в которых используются различные анимационные и звуковые эффекты.

4 Microsoft Access – средство для создания и эксплуатации баз данных. Данная система управления базами данных позволяет быстро найти необходимые пользователю данные и представить их в удобном для анализа виде с помощью таблиц, графиков и отчетов.

5 Microsoft Outlook является системой работы с сообщениями, которая обеспечивает связь с другими пользователями. Она позволяет хранить информацию о контактах, адреса, телефоны, получать и отправлять сообщения по электронной почте.

К достоинствам Microsoft Office также относится интегрированность программ Excel, Word, Access друг с другом. Данные, созданные в разных приложениях, входящих в эту систему, легко импортируются и экспортируются из одного приложения в другое.

3.4 Инструментальное программное обеспечение

Инструментальное ПО предназначено для разработки новых программ и включает в себя системы программирования и инструментальные среды.

Системы программирования и инструментальные среды – это совокуп-

ности языковых и программных средств, обеспечивающих разработку и подготовку к выполнению программ на определенном языке программирования.

Язык программирования – это система обозначений, используемых для описания данных и алгоритмов их обработки.

Любая программа есть последовательность предписаний, выполнив которые можно за конечное число шагов перейти от исходных данных к результату. В зависимости от степени детализации предписаний обычно определяется уровень языка программирования – чем меньше детализация, тем выше уровень языка.

По этому критерию можно выделить следующие классы языков программирования:

- 1) машинные языки – языки программирования, воспринимаемые аппаратной частью компьютера (машинные коды);
- 2) машинно-ориентированные языки – языки программирования, которые отражают структуру конкретного типа компьютера (ассемблеры);
- 3) машиннонезависимые языки программирования (языки высокого уровня):
 - процедурные (алгоритмические) – не зависят от архитектуры компьютера, отражают структуру алгоритма решения задачи (Pascal, C, Basic и др.);
 - проблемно-ориентированные – для решения задач определенного класса (языки документирования, управления, баз данных, машинной графики и др.);
 - объектно-ориентированные – поддерживают понятие объектов, их свойств и методов обработки.

Языки высокого уровня были разработаны для того, чтобы освободить программиста от учета технических особенностей конкретных компьютеров. Такие языки объединяют многие машинные команды в одну команду программы (оператор), при этом экранируется (скрывается) от программиста детализация действий компьютера (например, какие регистры памяти и для чего использовать).

3.4.1 Трансляция программ: основные понятия, этапы трансляции

Внутренний (машинный) язык любой ЭВМ – цифровой, данные и команды записываются в нем в двоичных кодах, в виде последовательности нулей и единиц. ЭВМ может выполнить только программы, представленные в виде машинного кода. Поэтому программы, разработанные на языках программирования, предварительно должны быть переведены на машинный язык.

Процесс преобразования исходного текста программы в машинный код называется *трансляцией* программы, а специальная программа, обеспечивающая перевод текстов программ, называется *транслятором*.

Трансляторы бывают двух видов:

- *компилятор* – читает, анализирует и переводит на машинный код всю программу, но не выполняет ее;
- *интерпретатор* – читает один оператор программы, анализирует его и сразу выполняет, после чего переходит к обработке следующего оператора.

В процессе трансляции программы можно выделить три главных этапа:

1) этап лексического анализа позволяет распознать лексические конструкции (идентификаторы, служебные слова, разделители и т. д.), характерные для языка программирования, и выявить допущенные ошибки;

2) этап синтаксического анализа позволяет распознать синтаксические конструкции (операции и управляющие структуры) в тексте исходной программы и проверить синтаксическую правильность программы;

3) этап генерации кода результирующей программы.

Если исходная программа содержит хотя бы одну ошибку, то результатом работы транслятора будет сообщение об ошибке (как правило, с дополнительными пояснениями и указанием места ошибки в исходной программе).

После преобразования исходного кода программы в машинные коды выполняется этап компоновки, на котором с помощью специальной программы - компоновщика (редактора связей) – к коду программы подключаются все необходимые библиотечные функции и структуры. В результате строится загрузочный модуль программы, пригодный к выполнению (рисунок 3.2).



Рисунок 3.2 – Схема процесса создания исполняемого файла программы

3.4.2 Структура системы программирования

1) Основными компонентами системы программирования являются:

2) специализированный текстовый редактор (редактор программ);

3) транслятор;

4) редактор связей (компоновщик) – предназначен для связывания между собой результирующих файлов, порождаемых компилятором, и файлов библиотек, входящих в состав системы программирования. Результатом его работы является единый файл (часто называемый «исполняемым файлом»), который содержит весь текст результирующей программы на языке машинных кодов;

5) отладчик – позволяет анализировать работу программы во время ее выполнения;

6) библиотеки подпрограмм (функций), в частности системная библиотека, содержащая стандартные функции входного языка программирования;

7) справочная система, содержащая описание языка программирования и правил составления программ на его основе.

Инструментальные среды помимо перечисленных компонентов включают в себя средства визуального проектирования пользовательского интерфейса (для создания системы меню, экранных форм и т. п.) и дополнительные вспомогательные программы (система управления проектом программного комплекса, генераторы приложений, специальные технологии работы с базами данных и

т. д.). Основное назначение инструментальных сред – автоматизация создания кодов программ, обеспечивающих графический интерфейс пользователя.

Типичными представителями систем программирования являются, например, известные пакеты Turbo-Pascal, Turbo-C, Turbo-Prolog фирмы Borland для IBM-совместимых компьютеров. Примерами инструментальных сред являются Delphi, Visual Basic, Visual C++ и другие.

3.5 РАСПРОСТРАНЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

По способам распространения ПО делят на коммерческое, условно-бесплатное и свободно распространяемое.

Коммерческое ПО создается и распространяется только на платной основе. Его легальное использование допустимо только в том случае, если пользователь приобрел право на использование соответствующего программного продукта. Условно-бесплатное ПО может использоваться бесплатно в течение установленного разработчиком срока. По истечении этого срока пользователь должен оплатить право на его применение или отказаться от использования программы. Свободно распространяемое ПО может использоваться бесплатно в течение любого срока.

Порядок использования программ регулируется лицензиями, в которых определяются права автора и пользователя программы. Существуют различные виды лицензий. Как правило, лицензионное соглашение на использование коммерческого ПО разрешает установить и использовать программный продукт только оговоренном лицензией числе компьютеров. Лицензионные соглашения на использование свободно распространяемого ПО обычно позволяют создавать любое число его копий, но запрещают коммерческое использование программы без ведома автора.

Правовая основа охраны программных продуктов как объектов интеллектуальной собственности заложена в двух законодательных актах РФ: Законе РФ № 3523-1 от 23 сентября 1992 г. «О правовой охране программ для электронно-вычислительных машин и баз данных» и Законе РФ № 5352-1 от 9 июля 1993 г. (с последующими изменениями) «Об авторском праве и смежных правах», которые определяют, что программы для ЭВМ и базы данных относятся к объектам авторского права. Статья 146 УК РФ предусматривает ответственность за незаконное использование объектов авторского права в виде крупных штрафов или лишения свободы на срок до двух, а в особо тяжких случаях до пяти лет.

3.6 Контрольные вопросы

- 1 Что называют программным обеспечением компьютера?
- 2 Что такое системное программное обеспечение? Для чего оно предназначено?
- 3 Что такое драйвер?
- 4 Что называется операционной системой?
- 5 Какие функции выполняет операционная система?

- 6 Назовите основные функциональные модули операционной системы.
- 7 Что входит в ядро операционной системы?
- 8 Что называется файлом?
- 9 В чем сходство и различие каталогов и файлов?
- 10 Назовите основные характеристики файла.
- 11 Что называют файловой системой?
- 12 Какие функции выполняет файловая система?
- 13 Назовите основные типы сервисных программ.
- 14 Для чего предназначено прикладное ПО?
- 15 Что относится к прикладному ПО общего назначения?
- 16 Назовите виды проблемно-ориентированного ПО.
- 17 Для чего предназначено инструментальное ПО?
- 18 Что такое язык программирования? Назовите основные классы языков программирования.
- 19 В чем различие между компилятором и интерпретатором?
- 20 Назовите составные части систем программирования.

Список источников

- 1 Гордеев А. В. Операционные системы : учебник. – 2-е изд. – Санкт-Петербург : Питер, 2004. – 416 с.
- 2 Информатика : базовый курс : учеб. пособие для студ. вузов / под ред. С. В. Симоновича. – 2-е изд. – Санкт-Петербург : Питер, 2004. – 639 с.
- 3 Олифер В. Г., Олифер Н. А. Сетевые операционные системы : учебник для вузов. – Санкт-Петербург : Питер, 2003. – 544 с.
- 4 Таненбаум Э. Современные операционные системы. – Санкт-Петербург : Питер, 2004. – 1040 с.

4 КОМПЬЮТЕРНЫЕ СЕТИ, ИНТЕРНЕТ И МУЛЬТИМЕДИА-ТЕХНОЛОГИИ

4.1 ПОНЯТИЕ КОМПЬЮТЕРНОЙ СЕТИ [5]

Телекоммуникационная вычислительная сеть (ТВС) – это сеть обмена и распределенной обработки информации, образуемая средствами связи и множеством взаимосвязанных абонентских систем; средства передачи и обработки информации ориентированы в ней на коллективное использование общих ресурсов.

Абонентская система (АС) – это совокупность ЭВМ, программного обеспечения, периферийного оборудования и средств связи с коммуникационной подсетью вычислительной сети.

Коммуникационная подсеть, или телекоммуникационная система (ТКС), представляет собой совокупность физической среды передачи информации и аппаратных и программных средств, обеспечивающих взаимодействие АС.

Компьютерной сетью называется два и более компьютера, взаимодействующих через среду передачу данных. Под средой передачи данных будем понимать кабельную систему (например, обычный телефонный провод, оптический волоконный кабель) и различные типы беспроводной связи (инфракрасное излучение, лазер и специальные виды радиопередачи).

Компьютеры, входящие в сеть, могут совместно использовать данные, принтеры, факсимильные аппараты, модемы и другие устройства.

Для сетей принципиальное значение имеют следующие обстоятельства:

- компьютеры, находящиеся в составе разных абонентских систем сети связываются между собой автоматически;

- каждый компьютер сети должен быть приспособлен как для работы в автономном режиме под управлением своей операционной системы (ОС), так и для работы в качестве составного звена сети. ТВС могут работать в различных режимах: обмена данными между АС, запроса и выдачи информации, сбора информации, пакетной обработки данных по запросам пользователей с удаленных терминалов, в диалоговых режимах.

По сравнению с аналогичной по вычислительной мощности совокупностью автономно работающих компьютеров сеть имеет ряд преимуществ:

- обеспечение распределенной и параллельной обработки данных многими ЭВМ;

- возможность распределенного хранения данных, т.е. размещения их в памяти различных ЭВМ;

- возможность обмена большими массивами информации между ЭВМ, удаленными друг от друга на значительные расстояния;

- повышение эффективности использования средств вычислительной техники за счет более интенсивной и равномерной их загрузки, а также надежности обслуживания запросов пользователей;

- предоставление большего перечня услуг, в том числе таких, как элек-

тронная почта (ЭП), телеконференции, электронные доски объявлений (ЭДО), дистанционное обучение;

- коллективное использование дорогостоящих программных и аппаратных ресурсов;
- удешевление работ по обслуживанию технических и программных средств.

Характеризуя возможности той или иной ТВС, следует оценивать ее аппаратное, информационное и программное обеспечение.

Аппаратное обеспечение составляют ЭВМ различных типов и аппаратура связи. Основные требования к ЭВМ сетей – это универсальность, т.е. возможность выполнения большого круга задач пользователей, и модульность, обеспечивающая возможность изменения конфигурации ЭВМ. В сетях в зависимости от их назначения используются ЭВМ широкого диапазона: от суперЭВМ до ПЭВМ.

Информационное обеспечение сети представляет собой единый информационный фонд, ориентированный на решаемые в сети задачи и содержащий данные для общего и индивидуального применения, доступные для пользователей сети.

Программное обеспечение (ПО) вычислительных сетей характеризуется большим многообразием как по своему составу, так и по выполняемым функциям. Оно автоматизирует процессы программирования задач обработки информации, осуществляет планирование и организацию коллективного доступа к телекоммуникационным, вычислительным и информационным ресурсам сети, динамическое распределение и перераспределение этих ресурсов с целью повышения оперативности и надежности удовлетворения запросов пользователей и т.д.

Выделяются следующие группы ПО сетей:

- общесетевое ПО, образуемое распределенной операционной системой сети и программными средствами для технического обслуживания сети (это контролирующие программы для контроля работоспособности элементов и звеньев сети и диагностические программы для локализации неисправностей);
- специальное ПО, представленное прикладными программными средствами, отражающими специфику предметной области пользователей при реализации своих задач;

- базовое программное обеспечение ЭВМ абонентских систем, включающее операционные системы ЭВМ, системы автоматизации программирования, контролирующие и диагностические программы.

Распределенная операционная система (РОС) сети управляет работой сети во всех ее режимах, обеспечивает реализацию запросов пользователей, координирует функционирование звеньев сети. РОС представляет собой систему программных средств, реализующих процессы взаимодействия АС.

Набор управляющих и обслуживающих программ РОС обеспечивает:

- удовлетворение запросов пользователей по использованию сетевых ресурсов, т.е. возможность доступа отдельных прикладных программ к ресурсам сети;

- синхронизацию работы пользовательских программ при их одновременном обращении к одному и тому же сетевому ресурсу;
- удаленный ввод заданий с любой АС сети и их выполнение в любой другой АС сети в пакетном или оперативном режиме;
- обмен файлами между АС сети;
- функционирование разнообразных сервисных служб;
- защиту информации и ресурсов сети от несанкционированного доступа, т.е. реализацию функций службы безопасности сети;
- возможность мониторинга сети.

С помощью РОС осуществляется планирование использования общесетевых ресурсов: установление сроков и очередности получения и выдачи информации пользователям, распределение решаемых задач по ЭВМ сети, присвоение приоритетов задачам, изменение конфигурации сети и т.д.

Основным показателем эффективности организации вычислительного процесса в сети, планирования использования общесетевых ресурсов является время решения комплекса задач.

Основное назначение компьютерных сетей – совместное использование ресурсов (удобный и надёжный доступ пользователей к общесетевым ресурсам) и осуществление интерактивной связи как внутри одной фирмы, так и за ее пределами (организация коллективного использования ресурсов). Ресурсы – это данные, приложения и периферийные устройства, такие, как внешний дискет, принтер, мышь, модем и джойстик. Понятие интерактивной связи компьютеров подразумевает обмен сообщениями в реальном режиме времени.

С появлением сетей решаются две проблемы:

- обеспечение неограниченного доступа к ЭВМ пользователей независимо от их расположения;
- возможность оперативного перемещения больших массивов информации на любые расстояния.

4.2 КЛАССИФИКАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ [4, 5]

Компьютерные сети отличаются сложностью и сферой деятельности. Вследствие этого они классифицируются различными способами. Однако наиболее распространенный способ оценки сетей основывается на размерах географической площади, покрываемой сетью. Первоначально компьютерные сети были небольшими и объединяли до десяти компьютеров и один принтер. Технология ограничивала размеры сети, в том числе количество компьютеров в сети и ее физическую длину. Например, в начале 1980-х годов наиболее популярный тип сетей состоял не более чем из 30 компьютеров, а длина ее кабеля не превышала 185 метров. Такие сети легко располагались в пределах одного этажа здания или небольшой организации. Для маленьких фирм подобная конфигурация подходит и сегодня. Эти сети называются локальными вычислительными сетями (ЛВС) (LAN, Local Area Network). Самые первые

типы локальных сетей не могли соответствовать потребностям крупных предприятий. Вследствие этого возникла необходимость в расширении локальных сетей. Сегодня, когда географические рамки сетей раздвигаются, чтобы соединить пользователей из разных городов и государств, ЛВС превращаются в глобальную вычислительную сеть (ГВС) (WAN, Wide Area Network), а количество компьютеров в сети практически не ограничено.

Приведем основные классификации сетей. Итак, сети классифицируются:

1 По степени территориального рассредоточения:

- глобальные (WAN);
- региональные (MAN);
- локальные (LAN).

2 По функциональному назначению:

- информационные;
- вычислительные;
- информационно-вычислительные.

Основной функцией вычислительных сетей является обработка информации, информационных – получение справочных данных по запросам пользователей, в смешанных – выполнение вычислительных и информационных функций.

3 По типу среды передачи:

- проводные сети, то есть сети, каналы связи которых построены с использованием медных или оптических кабелей;
- беспроводные сети, то есть сети, в которых для связи используются беспроводные каналы связи, например радио, СВЧ, инфракрасные или лазерные каналы.

Тип среды передачи влияет на технологию компьютерной сети, так как ее протоколы должны учитывать скорость и надежность соединения, обеспечиваемого каналом, а также частоту искажения в нем битов информации. Качество канала связи зависит от многих факторов, но наиболее кардинально на него влияет выбор проводной или беспроводной среды.

4 По способу управления ТВС:

- с централизованным управлением;
- с децентрализованным управлением;
- со смешанным управлением.

В централизованных сетях имеется один или несколько управляющих органов, децентрализованных – каждая АС имеет средства для управления сетью, а в сетях со смешанным управлением реализованы принципы централизованного и децентрализованного управления.

5 По технологии передачи:

- широковещательные;
- с маршрутизацией информации.

В широковещательных сетях взаимодействие АС производится следующим образом. Существует единый канал связи, совместно используемый всеми машинами в сети. Сообщения, называемые в некоторых случаях пакетами, посылаемые одной машиной, получают всеми машинами сети. Поле адреса данного

пакета указывает, кому предназначается данная информация. При получении пакета машина сверяет его адрес со своим собственным, если адрес совпадает, то пакет обрабатывается. Иначе он отбрасывается. Сети с маршрутизацией информации, напротив, состоят из большого количества соединенных между собой пар машин, возможно, соединенных не напрямую, а опосредованно. Тогда существует несколько возможных путей от источника к получателю. Поэтому с помощью коммуникационных систем сети решается задача выбора оптимального (например, кратчайшего по времени доставки кадра) маршрута.

6 По топологии сети:

- с базовой топологией (шина, звезда, кольцо);
- со смешанной топологией.

Под топологией сети будем понимать граф, вершинам которого соответствуют конечные узлы сети (например, компьютеры) и коммуникационное оборудование (например, маршрутизаторы), а ребрам – физические или информационные связи между вершинами [4, с. 55].

Среди множества возможных конфигураций различают полносвязные и неполносвязные.

Полносвязная топология (рисунок 4.1 а) соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными. Несмотря на логическую простоту, этот вариант оказывается громоздким и неэффективным. В этом случае каждый компьютер в сети должен иметь большое количество коммуникационных портов, чтобы установить соединение с каждым из остальных компьютеров сети. Для каждой пары компьютеров должна быть выделена отдельная физическая линия связи. Этот вид топологии чаще всего используется в сетях, объединяющих небольшое количество компьютеров.

Ячеистая топология получается из полносвязной путем удаления некоторых связей (рисунок 4.1 б). Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для крупных сетей.

В сетях с *кольцевой топологией* (рисунок 4.1 в) данные передаются по кольцу от одного компьютера к другому.

Главным достоинством кольца является то, что оно обеспечивает резервирование связей: любая пара узлов соединена здесь двумя путями – по часовой стрелке и против нее. Кольцо также представляет собой очень удобную конфигурацию для организации обратной связи – данные, сделав полный оборот, возвращаются к узлу-источнику. Это свойство кольца часто используется для тестирования связности сети и поиска узла, работающего некорректно. Из недостатков этой топологии нужно отметить необходимость реализации определенных мер, чтобы в случае выхода из строя или отключения какого-либо компьютера канал связи между остальными узлами кольца не прерывался.

Звездообразная топология (рисунок 4.1 г) образуется в случае, когда каждый компьютер подключается непосредственно к общему центральному устройству, называемому концентратором. В функции концентратора вхо-

дит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. В качестве концентратора может выступать как универсальный компьютер, так и специализированное устройство. Главным достоинством такой топологии является ее высокая надежность: при разрыве связи с одним компьютером работоспособность всей сети не нарушается. К недостаткам звездообразной топологии относится более высокая стоимость сетевого оборудования из-за необходимости приобретения специализированного центрального устройства. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора.

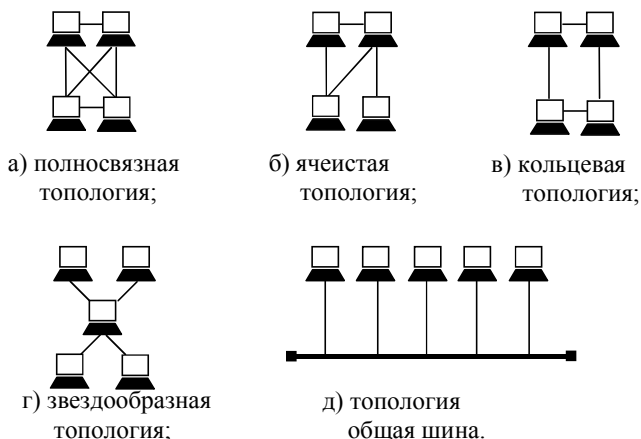


Рисунок 4.1 – Физические топологии сетей

Особым частным случаем звезды является *общая шина* (рисунок 4.1 д). Здесь в качестве центрального элемента выступает пассивный кабель, к которому подключается несколько компьютеров (такую же топологию имеют многие сети, использующие беспроводную связь – роль общей шины здесь играет общая радиосреда). Передаваемая информация распространяется по кабелю и доступна одновременно всем компьютерам, присоединенным к этому кабелю. Основными преимуществами такой схемы являются ее дешевизна и простота присоединения новых узлов к сети, а недостатками – низкая надежность (любой дефект кабеля полностью парализует всю сеть) и невысокая производительность (в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность делится здесь между всеми узлами сети).

Глобальные и региональные сети, как и локальные, могут быть однородными (гомогенными), в которых применяются программно-совместимые ЭВМ, и неоднородными (гетерогенными), включающими программно-несовместимые ЭВМ. Однако, учитывая протяженность ГВС и РВС и большое количество используемых в них ЭВМ, такие сети чаще бывают неоднородными.

4.3 ЛОГИЧЕСКАЯ ТОПОЛОГИЯ И ОСНОВНОЕ КОММУНИКАЦИОННОЕ ОБОРУДОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ [3]

Определение, данное в предыдущем разделе – это определение физической топологии, которая определяет порядок соединения компьютеров. Однако порядок их соединения, в общем случае, не определяет порядок «передвижения» информации по этим соединениям. Например, на рисунке 4.2 а приведена топология «общая шина», однако информация в такой сети перемещается по кольцу (пунктирные линии).

Еще пример: при построении сетей достаточно часто требуется изолировать трафик между различными участками одной сети (рисунок 4.2 б). Например, есть два отдела (А и В), компьютеры которых внутри отделов достаточно интенсивно обмениваются данными, а вовне отправляют ее значительно реже. Если не предусмотреть использование специализированного устройства, изолирующего трафик, то скорость обмена информацией между компьютерами этих отделов будет достаточно низкой вследствие того, что они используют общую среду передачи данных (топология общая шина).

Во всех этих случаях применяется специализированное оборудование, основных представителей которых мы рассмотрим в этом разделе, но, прежде всего, отметим, что логическая топология определяет пути движения информации по физическим носителям, она разделяет общую среду передачи данных на логические сегменты, устраняя столкновения (коллизии) данных в вычислительных сетях. Логические сегменты (подсети) могут работать автономно и, при необходимости, компьютеры из разных подсетей могут обмениваться информацией.

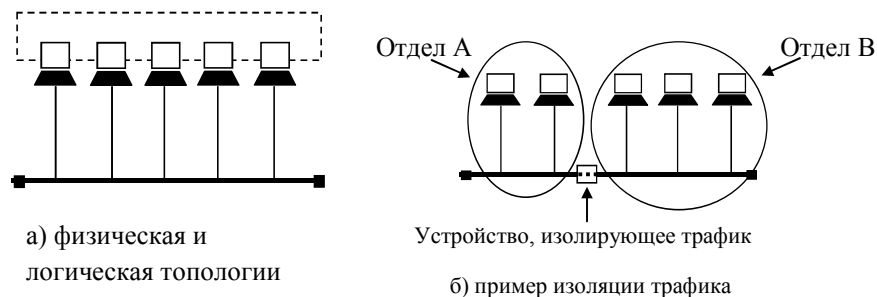


Рисунок 4.2 – Примеры логической топологии

Перечислим основные устройства, которые используются при построении сети. Начнем мы с сетевой карты.

Сетевой адаптер (сетевая карта) – это устройство двунаправленного обмена данными между компьютером и средой передачи данных вычислительной сети. Кроме организации обмена данными между компьютером и вычислительной сетью, сетевой адаптер выполняет буферизацию (временное хранение

ние данных) и функцию сопряжения компьютера с сетевым кабелем.

Для соединения отдельных компьютеров и коммуникационного оборудования в компьютерных сетях чаще всего применяются витая пара, коаксиальный и/или оптический кабель.

Промежуточное коммуникационное оборудование вычислительных сетей используется для усиления и преобразования сигналов, для объединения ПК в физические сегменты, для разделения вычислительных сетей на подсети (логические сегменты) с целью увеличения производительности сети, а также для объединения подсетей (сегментов) и сетей в единую вычислительную сеть. Перечислим основные устройства, которые используются для этих целей и дадим их краткую характеристику.

Повторители (repeaters) – это аппаратные устройства, предназначенные для восстановления и усиления сигналов в вычислительных сетях с целью увеличения их длины.

Концентраторы (hubs) и *коммутаторы (switches)* служат для объединения нескольких компьютеров в требуемую конфигурацию локальной вычислительной сети.

Концентраторы – это аппаратные устройства множественного доступа, которые объединяют в одной точке отдельные физические отрезки кабеля, образуют общую среду передачи данных или физические сегменты сети.

Коммутаторы – это программно-аппаратные устройства, которые делят общую среду передачи данных на логические сегменты. Логический сегмент образуется путем объединения нескольких физических сегментов с помощью концентраторов. Каждый логический сегмент подключается к отдельному порту коммутатора.

Концентраторы являются средством физической структуризации вычислительной сети, так как разбивают сеть на сегменты. Коммутаторы предназначены для логической структуризации вычислительной сети, так как разделяют общую среду передачи данных на логические сегменты и тем самым устраняют столкновения (коллизии).

Для соединения подсетей (логических сегментов) и различных вычислительных сетей между собой в качестве межсетевых интерфейсов применяются *коммутаторы, мосты (bridges), маршрутизаторы (routers)* и *шлюзы (gateways)*.

Мосты – это программно-аппаратные устройства, которые обеспечивают соединение нескольких локальных сетей между собой или несколько частей одной и той же сети, работающих с разными протоколами. Мосты предназначены для логической структуризации сети или для соединения в основном идентичных сетей, имеющих некоторые физические различия. Мост изолирует трафик одной части сети от трафика другой части, повышая общую производительность передачи данных.

Маршрутизаторы – это коммуникационное оборудование, которое обеспечивает выбор маршрута передачи данных между несколькими сетями, име-

ющими различную архитектуру или протоколы. Маршрутизаторы применяют только для связи однородных сетей и в разветвленных сетях, имеющих несколько параллельных маршрутов.

Шлюзы – это коммуникационное оборудование, служащее для объединения разнородных сетей с различными протоколами обмена данными. Шлюзы полностью преобразовывают весь поток данных, включая коды, форматы, методы управления и т.д.

4.4 ПОЯВЛЕНИЕ И РАЗВИТИЕ СЕТИ ИНТЕРНЕТ [8]

Прообраз сети Интернет был создан в конце шестидесятых годов по заказу Министерства обороны США. Разработка такой системы была поручена нескольким крупным университетам Америки. Компьютерную сеть в проекте назвали ARPANET (англ. Advanced Research Projects Agency Network), и уже в 1969 году сеть связала четыре университета: Калифорнийский, Стэндфордский, а так же университеты Калифорнии и Санта-Барбары. Все работы получали финансирование из средств Министерства обороны США. Позже сеть ARPANET была задействована учёными из разных областей науки. При этом Министерство обороны поставило условие, чтобы сеть продолжала работать при уничтожении ее части, поэтому повышенная надежность Интернета была заложена при его создании.

После первой успешной передачи данных в сети ARPANET следующим значимым этапом стала разработка в 1971 году первой программы для отправки электронной почты по сети.

К 1973 году в состав сети были включены первые зарубежные организации из Великобритании и Норвегии через трансатлантический телефонный кабель. С этого момента сеть стала считаться международной.

В 70-х годах прошлого века основным предназначением сети была рассылка электронной почты. В то же время появляются первые почтовые рассылки, различные доски объявлений и новостные группы. Однако во взаимодействии с другими сетями, построенными на других стандартах, были большие проблемы, которые решил переход на общий протокол TCP/IP (на данный момент – самый популярный протокол в сети). Этот переход состоялся 1 января 1983 года. Именно в этом году сеть ARPANET закрепила за собой термин «Интернет».

Следующим этапом развития была разработка системы доменных имён (англ. Domain Name System, DNS), которая состоялась в 1984 году.

В 1988 году был анонсирован протокол мгновенной передачи текстовых сообщений Internet Relay Chat (IRC), вследствие этого в Интернете стало возможным «живое» общение в чате в реальном времени.

В 1989 году знаменитый британский учёный Тим Бернерс-Ли предлагает концепцию Всемирной паутины. Он так же за два последующих года разрабатывает протокол HTTP, язык гипертекстовой разметки HTML и идентификаторы URI.

В 1990 году состоялось первое подключение к сети Интернет по телефонной линии (Dialup access – «дозвон»).

1991 год ознаменовался общедоступностью Всемирной паутины в Интернете.

В 1995 году роль маршрутизации всего сетевого трафика Интернета возложили на себя сетевые провайдеры. В этом же году был образован Консорциум всемирной паутины (W3C), призванный упорядочить веб-стандарты.

С 1996 году Всемирная паутина (WWW) почти полностью подменяет собой понятие «Интернет», и обгоняет по трафику протокол пересылки файлов FTP.

В 1990-е годы произошло массовое объединение большинства существовавших сетей под флагом Интернет (хотя такие сети как Фидонет так и остались обособленными). Открытость технических стандартов во многом способствовало быстрому росту сети. К 1997 году в Интернете насчитывалось около 10 млн. компьютеров и более 1 млн доменных имён.

В России Интернет появился в начале 90-х годов прошлого века. Подавляющее большинство пользователей могло лишь принимать и получать электронные письма, но не могло подключаться к Интернету и использовать все его возможности. Бурный рост числа пользователей Интернета в России начался в 1996 году. Сегодня в нашей стране Интернет превратился из диковинки в повседневный инструмент, подтверждением чему является бурное развитие русской части Интернета. Если несколько лет назад почти вся информация в сети приводилась на английском языке и предназначалась в основном для иностранцев, то сегодня в основном поставщики информации ориентируются на отечественных пользователей, и в Интернете можно найти самую разнообразную информацию на русском языке.

Отличительной особенностью Интернета является высокая надежность. При выходе из строя части компьютеров и линий связи сеть будет продолжать функционировать. Такая надежность обеспечивается тем, что в Интернете нет единого центра управления. Если выходят из строя некоторые линии связи или компьютеры, то сообщения могут быть переданы по другим линиям связи, так как всегда имеется несколько путей передачи информации.

Как и любая другая компьютерная сеть, Интернет состоит из множества компьютеров, соединенных между собой линиями связи. Пользователи Интернета подключаются к сети через компьютеры специальных организаций, которые называются поставщиками услуг Интернета. К сети могут быть подключены как отдельный компьютер, так и локальная сеть. В последнем случае можно считать, что к Интернету подключены все компьютеры данной локальной сети, хотя линией связи с Интернетом соединен только один компьютер. Соединение может быть постоянным или временным. Поставщики услуг Интернета имеют множество линий для подключения пользователей и высокоскоростные линии для связи с остальной частью Интернета. Часто мелкие поставщики подключены к более крупным, которые, в свою очередь, подключены к другим

поставщикам. Все организации, соединенные друг с другом самыми скоростными линиями связи, образуют базовую часть сети – Backbon. Если поставщик подключен непосредственно к Backbon, то скорость передачи информации будет максимальной.

В действительности разница между пользователями и поставщиками услуг Интернета достаточно условна. Любой человек, подключивший свой компьютер или свою локальную вычислительную сеть к Интернету и установивший необходимые программы, может предоставлять услуги подключения к сети другим пользователям.

В общем случае Интернет осуществляет обмен информацией между любыми двумя компьютерами, подключенными к сети. Компьютеры, подключенные к Интернету, часто называют узлами Интернета, или сайтами. Узлы, установленные у поставщиков услуг Интернета, обеспечивают доступ пользователей к Интернету. Существуют также узлы, специализирующиеся на предоставлении информации. Например, многие фирмы создают сайты, с помощью которых они распространяют информацию о своих товарах и услугах.

4.5 ПОНЯТИЕ МУЛЬТИМЕДИА [2]

Слово «мультимедиа» произошло от латинских слов «мульти» – много и «медиа» – среда, носитель. В целом это можно перевести как «множественная среда». И действительно: мультимедиа-технологии позволяют объединить различные виды информации в одно целое.

Мультимедиа – совокупность программно-аппаратных средств, реализующих интегрированную обработку и представление информации в символьном, звуковом и зрительном виде [1].

Основными особенностями мультимедиа-технологий являются [7]:

- объединение многокомпонентной информационной среды (текста, звука, графики, фото, видео) в однородном цифровом представлении;
- обеспечение надежного (без искажений при копировании) и долговечного хранения (гарантийный срок хранения – десятки лет) больших объемов информации;
- простота переработки информации.

Примером мультимедиа может служить DVD-диск с каким-либо фильмом. Посмотрим какие виды информации он в себе содержит. Во-первых, раз это фильм, то там будет видеoinформация. Во-вторых, редкий фильм обходится без звуков и диалогов, поэтому там должна быть аудиоинформация. В-третьих, там могут быть субтитры, которые представляют собой текстовую информацию. И в процессе воспроизведения фильма мы будем получать все эти виды информации сразу: видеть происходящее на экране, слышать звук и читать субтитры. Другим примером являются мультимедийные интернет-сайты, получившие распространение с появлением скоростных каналов доступа в интернет.

Области применения мультимедиа достаточно разнообразны: информа-

ционная и рекламная деятельности; шоу-бизнес; компьютерные тренажёры; компьютерные игры; обучающие программы; энциклопедии.

Во всех этих областях требуется комплексное представление данных, поэтому наиболее подходящей для них технологией является как раз мультимедиа.

Рассмотрим теперь основные составляющие мультимедиа по отдельности.

4.6 АУДИОИНФОРМАЦИЯ [2]

Обычно аудиоинформация носит изначально аналоговый характер. Но для компьютерной обработки её необходимо перевести в цифровую форму.

Для такого преобразования с определенной частотой (*частотой дискретизации*) производятся измерения амплитуды звукового сигнала. Затем непрерывные значения амплитуды тоже переводятся в дискретную форму путем разбивки интервала возможных значений амплитуды на конечное число промежутков и заменой текущего значения амплитуды на ближайшее граничное значение какого-либо интервала.

Количество бит, требуемое для записи получаемых таким образом значений, называется *разрядностью отсчёта*.

Было доказано, что при дискретизации аналогового сигнала потеря информации не будет только в том случае, если наивысшая частота сигнала не превосходит *частоту Найквиста*, которая равна половине частоты дискретизации. Т.к. обычное человеческое ухо воспринимает звуки до 20 кГц, то, согласно этому правилу, частота дискретизации должна быть не менее 40 кГц. Поэтому стандартными значениями частоты дискретизации для записываемых устройств являются 44,1 кГц (частота дискретизации музыкального компакт-диска), 48, 88,2 и 96 кГц (музыкальный DVD-диск). Стандартные разрядности отсчёта – 16, 20, 24 бит.

Сочетание частоты дискретизации, разрядности отсчёта и количества используемых каналов называют *форматом цифрового звука*.

Полученные непосредственно в результате дискретизации цифровые данные уже могут быть обработаны с помощью компьютера, однако, в таком виде их неудобно хранить и передавать по сети из-за большого объема. Поэтому чаще всего аудиоинформацию хранят в сжатом виде, а при воспроизведении распаковывают.

Существует два вида систем сжатия: с использованием аппаратных средств и с применением только программных методов. Быстродействие первых обычно выше, но их применение связано с установкой дополнительных устройств. Вторые представляют собой набор специализированных программ для сжатия и воспроизведения файлов, но их качество и коэффициент сжатия ниже.

Большая часть наиболее распространённых алгоритмов сжатия основана на том, что из звукового сигнала удаляется информация, малозаметная для слуха человека, в результате чего восприятие звука практически не меняется. Такое кодирование относится к методам сжатия с потерями, когда из сжатого

сигнала уже невозможно точно восстановить исходную форму.

Для описания степени сжатия звукового сигнала используется понятие *битрейт* – скорость битового потока, с которой сжатая информация должна поступать в декодер при восстановлении звукового сигнала. Измеряется битрейт в килобитах в секунду и показывает, сколько места будет требоваться для хранения 1 секунды сжатого звука. Чем выше битрейт, тем выше качество распакованного звука, но тем больше места будет требоваться для хранения этого звука.

Одним из наиболее распространённых форматов хранения аудио является mp3 (сокращение от MPEG Layer3). Он позволяет кодировать звук с битрейтом от 8 до 320 Кбит/с. Обладает высокой степенью компактности при сохранении качества звучания. Используется для передачи аудио по сетевым каналам и для кодирования Audio CD.

Другие форматы [1]:

- WAV. Этот формат является метаформатом для данных любого типа. Способ кодирования аудиосигнала может быть каким угодно. Может содержать данные, не имеющие отношения к аудио. Часто используется для хранения несжатой аудиоинформации;

- VQF. Этот формат разработан японской компанией NTT. Основным достоинством этого формата является высокая степень сжатия. VQF-файл может занимать на 30% меньше места, чем mp3-файл при одинаковом качестве звука. Главный недостаток алгоритма – высокое потребление системных ресурсов;

- WMA. Создан на основе разработок формата VQF. Он стал первым алгоритмом сжатия, обеспечивающим качество 128 кБит/с mp3-файлов при цифровом потоке в 64 кБит/с. Одним из основных недостатков является низкая стойкость к ошибкам: если при хранении/передаче часть файла повреждается, то воспроизведение этого файла после места повреждения невозможно;

OGG Vorbis. Свободный от патентных ограничений формат, который поддерживает до 255 каналов и обеспечивает сравнимую с VQF и WMA степень сжатия. Также в OGG-файлы можно встраивать графические изображения и тексты, которые могут возникать по ходу воспроизведения.

Помимо этого, в связи со значительным увеличением пропускной способности компьютерных сетей и ёмкости магнитных дисков, всё чаще применяются форматы flac, ape, alac, обеспечивающие сжатие без потерь, хотя коэффициент сжатия у них значительно ниже. Но зато эти форматы обеспечивают более высокое качество звучания.

4.7 ВИДЕОИНФОРМАЦИЯ [2]

Многое из написанного выше про аудиоинформацию справедливо и в случае видеоинформации. Преобразование аналогового видеосигнала в цифровой производится почти по тем же принципам, что и при работе с аудиосигналами. Основная разница заключается в том, что отсчёты делаются гораздо реже (25

раз в секунду), а результатом отсчёта является целый кадр. При этом оцифрованное видео будет занимать значительно больше места, чем аудио, так что для него сжатие ещё более актуально.

Сжатие и распаковка как аудио, так и видеосигналов производится с помощью специальных программных модулей, называемых *кодеками* (КОдерами-ДЕКОдерами). Они могут отличаться различными алгоритмами сжатия и используемыми форматами сжатых файлов.

Кодеки принято идентифицировать четырехзначным кодом FourCC (например, «DIV3», «DIVX», «DX50», «XVID») [6].

Для корректного воспроизведения видеоинформации кодеки, соответствующие формату видеозаписи, должны быть установлены на компьютере. В случае, если необходимого кодека нет на компьютере, его можно найти в интернете, узнав предварительно его код FourCC (это можно сделать, например, с помощью некоторых видеопроигрывателей, которые отображают дополнительную информацию о файле с видеозаписью).

Стандарты MPEG:

- MPEG 1 предназначен для записи синхронизированных видеоизображений (обычно в формате 352×288) и звукового сопровождения на CD-ROM (VideoCD). Качество примерно соответствует обычному VHS-видео.

- MPEG 2 поддерживает более высокие разрешения (до 720×576), позволяя записывать полноэкранные фильмы студийного качества. Этот формат используется в видеодисках DVD. Кроме того, сигнал, сжатый в соответствии с этим стандартом, транслируется через телевизионные спутники.

- MPEG 4 изначально создавался для использования в мультимедийных приложениях, использующих узкие каналы связи и не предназначался для хранения видео. По качеству изображения он занимает промежуточное место между MPEG 1 и MPEG 2. Характеризуется высокой степенью сжатия, но требует значительной вычислительной мощности от компьютера. Сейчас MPEG 4 зачастую используется для преобразования DVD-фильмов (формата MPEG 2) с целью значительного уменьшения их объема. Модификация MPEG 4 с таким предназначением носит название DivX.

Впоследствии стандарт MPEG 4 был расширен для обработки видеосигнала высокой четкости и в этом случае чаще всего называется H.264 или AVC/H.264.

Как уже отмечалось, мультимедиа – это средство комплексного представления информации. Для того, чтобы объединить и синхронизировать между собой видео, звук, субтитры и прочее, используются *контейнеры* – файлы специальных форматов, содержащие такую информацию. Тип контейнера можно определить по расширению соответствующего файла.

Один из самых распространенных является формат .avi (Audio Video Interleave – чередование видео и аудио). Более современным открытым форматом контейнера является .mkv (Matroska – Матрёшка), часто используемый в интернете. Примерами других контейнеров являются mov, mp4, 3gp, wmv, flv.

При этом следует понимать, что в контейнерах одного и того же типа мо-

гут использоваться разные кодеки. Поэтому вполне возможна ситуация, когда один файл с расширением .avi успешно воспроизводится, а другой с тем же расширением не может найти подходящий кодек.

4.8 Контрольные вопросы

- 1 Дайте определение телекоммуникационной вычислительной сети.
- 2 Что такое абонентская система?
- 3 Какая система называется телекоммуникационной?
- 4 Дайте определение компьютерной сети.
- 5 Перечислите преимущества использования компьютеров в сети по сравнению с автономными компьютерами. Какие недостатки вы можете указать при работе в сети?
- 6 Укажите основные группы сетевого программного обеспечения.
- 7 Перечислите основные функции распределенной операционной системы.
- 8 Перечислите основные классификации сетей.
- 9 Что понимается под топологией сети? Перечислите типовые физические топологии.
- 10 Что определяет логическая топология компьютерной сети?
- 11 Перечислите основные устройства, которые используются при построении сети.
- 12 Почему сеть Интернет является высоко надежной компьютерной сетью?
- 13 Что такое сайт?
- 14 Что такое мультимедиа?
- 15 В каких областях могут применяться мультимедиа-технологии?
- 16 Как производится оцифровка аудиосигнала?
- 17 Что такое битрейт и что он характеризует?
- 18 Что такое кодеки? Для чего они нужны?
- 19 Какие стандарты MPEG вы знаете? В чём их отличия друг от друга?
- 20 Для чего применяются контейнеры? Какие типы контейнеров вы знаете?

Список источников

- 1 Информационные технологии : учебник для вузов / О. Л. Голицына [и др.]. – Москва : Форум, 2008. – 544 с.
- 2 Гохберг Г. С., Зафиевский А. В., Короткин А. А. Информационные технологии : учебник для студ. учреждений сред. проф. образования. – Москва : Академия, 2014. – 240 с.
- 3 Коммуникационное оборудование вычислительных сетей. URL: http://www.lessons-tva.info/edu/telecom-loc/mlt6_2loc.html.
- 4 Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов. – Санкт-Петербург : Питер, 2010. – 944 с.

5 Пятибратов А. П. Вычислительные системы, сети и телекоммуникации. URL: <http://www.studfiles.ru/preview/718148/>.

6 Румянцева Е. Л., Слюсарь В. В. Информационные технологии : учеб. пособие / под ред. Л. Г. Гагариной. – Москва : Форум, 2007. – 256 с.

7 Советов Б. Я., Цехановский В. В. Информационные технологии : учебник для вузов. – 3 изд., стер. – Москва : Высшая школа, 2006. – 263 с.

8 Что такое Интернет? История и этапы развития. URL: <http://moolkin.ru/chto-takoe-internet-istoriya-i-etapy-razvitiya>.

5 ОСНОВЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

5.1 БАЗЫ ДАННЫХ. БАНКИ ДАННЫХ. СУБД

5.1.1 Понятие предметной области

Каждая информационная система в зависимости от ее назначения имеет дело с частью реального мира, которую принято называть *предметной областью* (ПО) системы. Предметная область может относиться к любому типу организаций: банк, университет, завод, магазин и т.д.

Предметная область информационной системы – это совокупность реальных объектов (сущностей), которые представляют интерес для пользователей.

Объект (сущность) – предмет, процесс или явление, о котором собирается информация, необходимая для решения задачи.

Объектом может быть человек, предмет, событие. Каждый объект характеризуется рядом основных свойств, которые принято называть *атрибутами*.

Атрибутом называется поименованная характеристика объекта.

Атрибут показывает, какая информация должна быть собрана об объекте. Например, объект – студент вуза, атрибуты – номер зачетной книжки, фамилия, имя, отчество, домашний адрес.

5.1.2 Модели данных

Понятие «данные» в концепции баз данных – это набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые другие факторы. Данные не обладают определенной структурой, данные становятся информацией тогда, когда пользователь задает им определенную структуру, то есть осознает их смысловое содержание. Поэтому центральным понятием в области баз данных является понятие модели данных.

В технологии баз данных модель данных – одно из фундаментальных понятий. *Модель данных* [14, с. 456-457] – это комбинация, по крайней мере, трех составляющих:

- 1) набора типов структур данных (являющихся блоками при построении БД);
- 2) набора операторов или правил вывода, которые могут быть применены к любым правильным примерам типов данных, перечисленных в пункте 1, чтобы находить, выдавать или преобразовывать информацию, содержащуюся в любых частях этих структур в любых комбинациях;
- 3) набор общих правил целостности, которые прямо или косвенно определяют множество непротиворечивых состояний базы данных, или множество изменений ее состояния, или и то и другое.

В настоящее время предложено более 30 моделей данных. В теории баз данных рассматриваются реляционные, сетевые, иерархические, модели дан-

ных «сущность-связь», бинарные модели, семантические сети, инфологические модели.

5.1.3 Базы данных. Классификация баз данных

В литературе понятие «база данных» трактуется по-разному, поэтому можно говорить о том, что «число определений базы данных сравнимо с числом существующих систем управления базами данных» [2, с. 44]. В частности:

- база данных – совокупность данных, организованных по определенным правилам, которые предусматривают общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ. [20; 4; 13];
- база данных – по законодательству РФ – объективная форма представления и организации совокупности данных, систематизированных таким образом, чтобы эти данные могли быть найдены и обработаны с помощью ЭВМ;
- база данных – поименованная совокупность взаимосвязанных данных, находящихся под управлением СУБД [7, с. 7].

База данных – именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области [23, с. 29].

Практически все базы данных обычно включают различные аппаратные и программные средства ЭВМ, используемые для хранения и манипулирования данными. Такие базы данных, как правило, опираются на определенную файловую систему, обеспечивающую выполнение простейших операций с данными.

Данные из одной базы данных взаимосвязаны, предназначены для одного или нескольких типов приложений и хранятся так, чтобы быть независимыми от использующих их программ. Базы данных в основном исполняют функции накопления и хранения информации об определенной предметной области, предоставляя ее для программ. Например, каталог любой библиотеки может служить примером неавтоматизированной базы данных. Он является системой, представляющей ряд свойств книг: фамилии авторов, название, год издания, издательство, шифр книги и т.п.

По используемой модели данных все базы данных могут быть разделены на три основных (или «базовых») типа: *сетевые*, *иерархические* и *реляционные*. Название типа определяется тем, с помощью каких структур представлена информация в базе данных (т.е. от модели данных).

Кратко охарактеризуем каждый из типов баз данных в зависимости от модели данных.

1 *Сетевые базы данных*. В них основной структурой являются *сети*, т. е. произвольные графы, в вершинах которых записана соответствующая информация, а ребра соответствуют связям между ними.

Граф – это математическая конструкция, состоящая из вершин и ребер. Каждое ребро соединяет две вершины. В сетевой структуре любая вершина может быть связана с любой другой (рисунок 5.1).

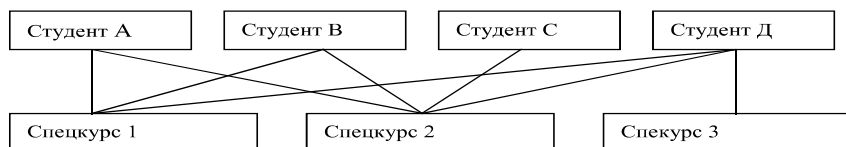


Рисунок 5.1 – Пример сетевой структуры

В сетевой структуре два уровня взаимосвязанных объектов: соотношение между ними – «многие ко многим».

Иерархические базы данных. В них основной структурой представления информации являются *деревья*.

Дерево определяют как связный граф, не имеющий циклов. Все вершины разбиты на уровни. На самом высшем уровне находится только одна вершина, которая называется корнем дерева. Любой уровень можно достигнуть лишь через корень дерева. Он соединяется ребрами со всеми вершинами, находящимися на втором уровне, и только с ними. Вершины второго уровня соединяются с вершинами третьего уровня ребрами так, что каждая вершина третьего уровня соединена только с одной вершиной второго уровня и т.д.

Графическое изображение дерева имеет следующий схематичный вид (рисунок 5.2):

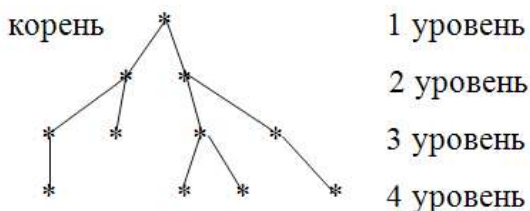


Рисунок 5.2 – Дерево

Пример иерархической структуры в виде дерева приведен на рисунок 5.3:

В дереве соотношение между верхними и нижними объектами имеет характер «один-ко-многим».

Если данные имеют естественную иерархическую структуру, то использование иерархических баз данных имеет преимущество – эффективное использование памяти компьютера и простота выполнения многих операций по преобразованию данных.

Реляционные базы данных. В реляционных базах данных основной структурой является *отношение*. Реляционная модель основана на понятии n -наборов декартова произведения полей. Название происходит от слова relation (отношение).

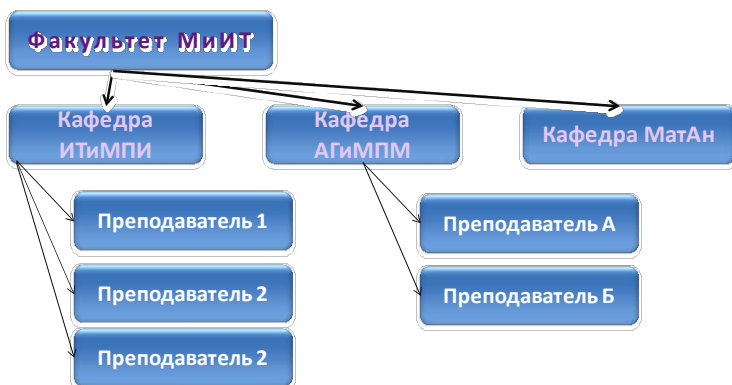


Рисунок 5.3 – Иерархическая структура «Факультет МИИТ»

Отношения изображаются в виде двумерных таблиц (таблица 5.1). В реляционной БД данные хранятся в *таблицах*. Каждая таблица представляет собой совокупность строк и столбцов, где строки соответствуют экземпляру объекта, конкретному событию или явлению, а столбцы – атрибутам (признакам, параметрам) объекта, события, явления.

Таблица 5.1 – Пример отношения

№	Ф. И. О. студента	Дата рождения	Домашний адрес
1	Иванов П. Г.	01.02.84	ул. Комсомольская, 37 - 23
2	Алексеева Т. Н.	12.03.81	ул. Ленина, 23 а - 12
3	Петров А. А.	01.02.85	ул. К. Мяготина, 22 - 1

Каждая строка описывает определенный объект (в нашем примере – студента). Каждый столбец содержит однородную информацию, т.е. все элементы этого столбца имеют одинаковую природу (тип). В терминах базы данных строку называют записью, а столбец – полем.

База данных состоит из записей; а записи делятся на поля. Запись является наименьшей единицей обмена данными между оперативной и внешней памятью; поле – наименьшей единицей обработки данных.

Вообще, при сравнении между собой трех «базовых» типов баз данных можно сделать следующие выводы:

- 1) обработка реляционных структур более проста, хотя и менее эффективна в некоторых случаях, чем структур других типов;
- 2) использование отношений во многом упрощает работу с базой данных, делая их доступными;
- 3) с помощью отношений можно выразить сетевые и иерархические структуры. Это делается за счет повторения одинаковых данных и в результате менее эффективно используется память компьютера. Например, если отноше-

ние «ученик посещает факультатив» обозначить 1, то сетевая структура (см. рисунок 5.1) примет вид таблицы и станет реляционной (таблица 5.2):

Таблица 5.2 - Реляционная таблица

	Спецкурс 1	Спецкурс 2	Спецкурс 3
Студент А	1	1	0
Студент В	1	1	0
Студент С	0	1	0
Студент Д	1	1	1

4) использование сетевых и иерархических структуры позволяет экономить память и в ряде случаев ускоряет обработку данных;

5) имеется возможность глубокого исследования функционирования и особенностей реляционных БД за счет развитого математического аппарата в теории отношений.

Базы данных различают по характеру хранимой информации на фактографические и документальные. В *фактографических* БД содержатся краткие сведения об объектах, представленных в строго определенном формате. *Документальная* БД – база данных, в запись которой отражает конкретный документ, содержит его библиографическое описание и, возможно, иную информацию о нем. Она содержит обширную информацию самого разного типа: текстовую, графическую, звуковую, мультимедийную.

5.1.4 Банки данных. Базы знаний. Отличие баз данных от баз знаний

С увеличением потока информации стали говорить о банках данных. Банк данных является компонентом автоматизированной системы и обычно состоит из нескольких баз данных, находящихся под общим управлением. Он создается в интересах решения задач по определенным направлениям науки и техники (банк данных по общественным наукам, банк изобретений и т.д.).

Итак, *банк данных* – основанная на технологии баз данных система программных, языковых, организационных и технических средств, предназначенных для централизованного накопления и коллективного использования [23 и 24].

Или 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.

В самом общем виде основные функции банка данных можно сформулировать следующим образом: адекватное информационное отображение предметной области, обеспечение хранения, обновления и выдачи необходимых данных пользователям.

Составными частями любого банка данных являются: базы данных, си-

стема управления базой данных (СУБД), администратор базы данных, прикладное программное обеспечение.

Возможности компьютеров и связанные с ними системы очень быстро развиваются. Одним из свидетельств этого является переход от обработки данных к оперированию со знаниями. Это, в свою очередь, приводит к появлению банков и баз знаний.

База знаний – организованная совокупность знаний, представленная в форме, которая допускает автоматическое или автоматизированное использование этих знаний с помощью ЭВМ [10, с. 137].

М. Бургин и Г. Степенко рассмотрели *отличия* знаний от данных:

1) знания показывают к чему относятся данные и что они означают, т.е. знания включают интерпретацию данных (Например, если в таблице 1 убрать первую строку, то получим данные);

2) знания всегда образуют некоторую систему [1].

5.1.5 Классификация систем управления базами данных

Общее управление в базах данных осуществляется специально предназначенной для этого системой управления базами данных (СУБД), состоящей из языковых, алгоритмических и программных средств. Под *системой управления базами данных* мы будем понимать совокупность программных и языковых средств, предназначенных для создания, ведения и использования информации, хранящейся в базе данных. Система управления базами данных является составной частью автоматизированного банка данных и обеспечивает работу прикладных программ с базой данных. *Главная цель СУБД* состоит в представлении пользователю возможности оперировать данными в близких ему терминах и понятиях, не связанных с конкретными способами хранения данных в компьютере.

СУБД 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. Необходимо отметить, что операция поиска является главной среди указанных выше.

Итак, система управления базами данных выполняет следующие *функции*:

- *описание структуры базы данных*. Подобными средствами являются *язык определения (описания) данных (ЯОД)* и *язык манипулирования данными (ЯМД)*;

- *создание, обновление и извлечение информации* из баз данных. Средством извлечения информации из баз данных является *язык обработки данных*.

- *защита (безопасность) и целостность данных*. Использование системы разрешается лишь пользователям, имеющим на это право (защита). При выполнении пользователями операций над данными поддерживается согласованность хранящихся данных (целостность).

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 других. Необходимо также отметить такие возможности СУБД, как обеспечение защиты и секретности данных, восстановление баз данных после сбоев, ведение учета работы с базами данных. Однако это является неполным перечнем того, что должна осуществлять СУБД для обеспечения интерфейса пользователей с базами данных и жизнеспособности всего автоматизированного банка данных.

Существующие системы управления базами данных делятся: [7]

1) *по типам используемых моделей данных* на реляционные, иерархические, сетевые и комбинированные;

2) *по сфере возможного применения* на универсальные и специализированные (обычно это проблемно-ориентированные СУБД). Рассмотрим универсальные СУБД, используемые для различных приложений. При настройке универсальных СУБД для конкретных приложений они должны обладать соответствующими средствами. Процесс настройки СУБД на конкретную область применения называется генерацией системы. К универсальным системам управления базами данных относятся, например, системы dBase, СУБД Paradox, Microsoft Access, Oracle;

3) *по языкам общения* на открытые, замкнутые и смешанные;

4) *по числу уровней в архитектуре* на одноуровневые, двухуровневые, трехуровневые;

5) *по выполняемым функциям* на информационные и операционные;

б) *в зависимости от расположения* на локальные и распределенные (удаленные) СУБД. Все части локальной СУБД располагаются на компьютере пользователя БД.

Отметим, что существуют и другие классификации СУБД.

Функционирование СУБД основано на введении двух уровней организации базы данных – логического и физического. Эти два уровня соответствуют двум аспектам организации данных: физическому с точки зрения хранения данных в памяти ЭВМ и логическому с точки зрения использования данных в прикладных приложениях. Описание логических организаций баз данных определяет взгляд пользователей на организацию данных в системе, которые отображают состояние некоторой предметной области. Необходимо отметить, что 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.

5.1.6 Классификация информационных систем

На основе баз данных создаются автоматизированные информационные системы. *Автоматизированная информационная система* (АИС) – система, реализующая автоматизированный сбор, обработку и манипулирование данными из базы данных и включающая технические средства обработки данных, программное обеспечение и соответствующий персонал [23, с. 123].

Часто понятия базы данных и информационной системы (ИС) используются в одном и том же смысле, однако между ними существуют определенные различия. Если БД в основном исполняет функции накопления и хранения информации, предоставляя ее для других систем, то ИС обслуживает непосредственно пользователей.

Различия между СУБД и информационно-поисковой системой заключаются в следующем: в СУБД мы формулируем запрос (например, в терминах реляционного исчисления или реляционной алгебры) и отыскиваем удовлетворяющие этому запросу кортежи (записи) в одном или нескольких отношениях (таблицах). В информационно-поисковой системе мы отыскиваем набор документов, связанных с интересующей нас темой или имеющих к ней отношение. Критерий поиска основывается на отношении, называемом *сходством*, и постепенно уточняется до достижения заданной цели. В то время как в СУБД мы имеем тождество между атрибутами запроса и атрибутами экземпляров типа записей, в информационно-поисковой системе обнаруженный документ может соответствовать атрибутам поиска лишь частично.

В основу классификации АИС положены наиболее существенные признаки, характеризующие возможности и особенности современных АИС.

Автоматизированные информационные системы делятся [13, с. 97]:

1) *по типу хранимых данных* на документальные и фактографические. Документальные системы служат для работы с документами на естественном языке – монографиями, публикациями в периодике и т.д. Наиболее распространенным видом документальных систем являются информационно-поисковые системы (ИПС). Характерной особенностью фактографических систем является то, что они оперируют фактическими сведениями, представленными в виде специальным образом организованных совокупностей форматизированных записей данных.

2) *по характеру обработки данных* на информационно-справочные и АИС обработки данных. Информационно-справочные системы (часто называемые запросно-ответными или справочными) выполняют поиск и вывод информации без ее обработки. АИС обработки данных сочетают в себе информационно-справочные системы с системой обработки данных.

3) *по степени интегрированности данных и автоматизации управления* на АИС на автономных файлах и банки данных.

4) *по степени распределенности* на локальные и распределенные.

О роли АИС и баз данных говорит то, что неотъемлемой частью информационной системы становятся базы данных, а в число основных функций входят задачи накопления информации в базе данных и выдача ее пользователям.

5.2 ОСНОВНЫЕ КОНЦЕПЦИИ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

5.2.1 Реляционная модель данных

Реляционная модель была предложена в 1969 году сотрудником фирмы IBM Е. Ф. Коддом (Dr. E. F. Codd). Первые основные концепции этой модели были опубликованы в 1970 году в статье «A Relational Model of Data for Large Shard Data Banks» (CACM, 1970, 13 № 6).

Набор средств для управления реляционными базами данных называется реляционной системой управления базами данных (РСУБД). Реляционная система управления базами данных может содержать утилиты, приложения, сервисы, библиотеки, средства создания приложений и другие компоненты.

Реляционная база данных представляет собой совокупность двумерных таблиц. Напомним, что любая таблица реляционной базы данных состоит из строк, называемых *записями*, и столбцов, называемых *полями*. Строки таблицы содержат сведения об объектах.

Возьмем, например, записную книжку с заметками о друзьях и знакомых. О каждом в книжке есть информация – фамилия, имя, адрес, номер телефона, хобби. Для облегчения чтения расположим всю информацию в виде таблицы, то есть по строкам и столбцам. Каждый тип информации занесем в свой собственный столбец – отдельно имя человека, отдельно его адрес и т.д. (таблица 5.3).

Таблица 5.3 – Пример реляционной таблицы

Фамилия, Имя, Отчество	Домашний адрес	Дата рождения	Телефон	Хобби
Андреев Олег Олегович	Ул. Зорге, 18-45	01.01.1980	5-46-28	Фотография
Иванов Иван Иванович	Ул. Тобольная, 12-45	23.03.1985		Рыбалка
Петров Петр Петрович	Ул. Красина, 45-1	08.08.1989	42-55-11	Филателия

5.2.2 Поле и запись таблицы

Каждый столбец в таблице должен содержать только определенный тип информации (например, дату рождения, или хобби Вашего друга, или домашний адрес и т.п.). Каждая *строка* таблицы содержит разнообразную (разного типа) информацию о человеке. Все данные, помещенные в одной строке, называют *записью*, каждый *элемент записи* – это *поле*. Таким образом, каждое поле содержит часть информации, находящейся на пересечении соответствующей строки и столбца. В таблице всевозможные значения одного типа в одном столбце называют *доменом*.

Поле содержит *значение* одного отдельного признака (фамилию знакомого, дату его рождения, домашний адрес, домашний телефон и т.д.).

Поле является элементом записи и представляет собой ячейку таблицы. У каждого столбца есть свое неповторимое имя, описывающее тот вид информации, который содержится в нем. Это имя называют «*именем поля* базы данных».

Запись – это полный набор данных об определённом объекте. Запись содержит совокупность значений признаков, описывающих *один* объект (например, человека, город, книгу, машину и т.д.) и представляет собой одну строку таблицы.

Запись – совокупность полей, соответствующих одному объекту.

Каждое поле имеет фиксированную длину, следовательно, и любая запись в таблице имеет фиксированную длину. Каждая запись характеризуется своим уникальным порядковым номером, т.е. ключевым полем.

Данные в реляционной таблице должны удовлетворять следующим правилам:

1 Каждое значение поля должно быть атомарным, т.е. не расчленимым (неделимым) на несколько значений;

2 Значения данных домена (в одном и том же столбце) должны принадлежать к одному и тому же типу данных, доступному для использования в данной СУБД;

3 Каждая запись в таблице уникальна, т.е. в таблице не существует двух записей с полностью совпадающим набором значений ее полей;

4 Каждое поле имеет уникальное имя;

5 Последовательность полей в таблице несущественна;

6 Последовательность записей в таблице несущественна.

5.2.3 Первичный и внешний ключи таблицы

Поскольку записи в таблице неупорядочены, то необходимо указать поле (или набор из нескольких полей) для уникальной идентификации каждой записи.

Поле или набор полей, которые однозначно идентифицируют (определяют) запись таблицы, называют *первичным ключом* (primary key).

Обычно ключом является поле или совокупность полей фиксированной длины. Каждому значению *первичного* или *основного* ключа соответствует одна и только одна запись. Первичный ключ любой таблицы обязан содержать уникальные непустые значения для каждой записи. Если первичный ключ состоит из нескольких полей, он называется *составным первичным ключом* (composite primary key).

Поле, указывающее на запись в другой таблице, связанную с данной записью, называется *внешним ключом* (foreign key).

Внешний ключ – это поле или набор полей, чьи значения совпадают с имеющимися значениями первичного ключа другой таблицы.

Подобное взаимоотношение между таблицами называется *связью* (relationship). Связь между двумя таблицами устанавливается путем присвоения значений внешнего ключа одной таблицы значениям первичного ключа другой. Таблица, содержащая внешний ключ, называется *второстепенной* или *detail-таблицей*, а таблица, содержащая первичный ключ, определяющий возможные значения внешнего ключа второстепенной таблицы, называется *главной* или *master-таблицей*.

Типичная реляционная база данных состоит из нескольких связанных таблиц. Приведем пример фрагмента базы данных «Деканат».

Таблица *Студенты* состоит из следующих полей¹ (таблица 5.4):

Таблица 5.4 – Таблица «Студенты»

Номер зачетной книжки PK	Фамилия, имя, отчество	Группа	Дата рождения	Домашний адрес	Телефон
-----------------------------	------------------------	--------	---------------	----------------	---------

Таблица *Успеваемость* состоит из следующих полей (таблица 5.5):

Таблица 5.5 – Таблица «Успеваемость»

Код отметки PK	Номер зачетной книжки FK	Код дисциплины FK	Семестр	Форма сдачи	Отметка	Дата сдачи	Код преподавателя FK
-------------------	-----------------------------	----------------------	---------	-------------	---------	------------	-------------------------

Таблица *Дисциплины* состоит из следующих полей (таблица 5.6):

Таблица 5.6 – Таблица «Дисциплины»

Код дисциплины PK	Название	Программа	Специальность
----------------------	----------	-----------	---------------

Таблица *Преподаватели* состоит из следующих полей (таблица 5.7):

Таблица 5.7 – Таблица «Преподаватели»

Код преподавателя PK	Фамилия, имя, отчество	Код кафедры FK	Должность	Ученая степень	Ученое звание
-------------------------	------------------------	-------------------	-----------	----------------	---------------

Таблица *Кафедры* состоит из следующих полей (таблица 5.8):

Таблица 5.8 - Таблица «Кафедры»

Код кафедры PK	Название	Код преподавателя (зав. кафедрой) FK
-------------------	----------	---

¹ В примере используются следующие обозначения: PK – первичный ключ таблицы; FK – внешний ключ таблицы.

Номер зачетной книжки у каждого студента индивидуален. Т.о. поле «Номер зачетной книжки» в таблице «Студенты» однозначно определяет студента, следовательно, это поле является ключевым полем таблицы «Студенты». Если требуется узнать успеваемость студента, следует найти значение идентификатора студента в поле «Номер зачетной книжки» таблицы «Успеваемость» и в найденных строках прочесть значение поля «Отметка». Иными словами, нужно связать две таблицы «Студенты» и «Успеваемость» по полю «Номер зачетной книжки». Внешним ключом таблицы «Успеваемость» является поле «Номер зачетной книжки».

Все информационные объекты предметной области связаны между собой. Соответствия, отношения, возникающие между объектами предметной области, называются *связями*. Связанные отношениями таблицы взаимодействуют по принципу *главная* (master), *подчиненная* (detail). Возможны следующие отношения между таблицами:

1 Отношение «*один-ко-многим*» (one-to-many relationship): одной записи из главной таблицы может соответствовать ноль, один или несколько записей подчиненной таблицы.

При связи «один-ко-многим» (обозначают 1:M) одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В, но каждый экземпляр объекта В связан не более чем с 1 экземпляром объекта А. Графически данное соответствие имеет вид, представленный на рисунке 5.4.



Рисунок 5.4 – Графическое изображение отношения 1:M

Пример 1. Установленный размер стипендии по результатам сдачи сессии может повторяться многократно для различных студентов, поэтому примером связи 1:M служит связь между информационными объектами *стипендия* и *результаты сессии*:

стипендия ↔↔ *результаты сессии*

Пример 2. Один студент за период обучения в вузе может сдать много экзаменов и зачетов по различным дисциплинам, поэтому между информационными объектами *студент* и *сессия* установлена связь 1:M:

стипендия ↔↔ *сессия*

2 Отношение «*один-к-одному*» (one-to-one relationship): одной записи из главной таблицы соответствует только одна запись из подчиненной. Связь «один-к-одному» (обозначают как 1:1) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует не более одного экземпляра информационного объекта В и наоборот (рисунок 5.5).



Рисунок 5.5 – Графическое изображение отношения 1:1

Пример 3. Студент может не «заработать» стипендию, получить обычную или одну из повышенных стипендий: *студент ↔ сессия*

Пример 4. Каждый студент может за период обучения может выполнить только одну дипломную работу, поэтому примером связи 1:1 может служить связь между информационными объектами *студент* и *дипломная*:

студент ↔ дипломная

Отношение «многие-ко-многим». Связь «многие-ко-многим» (M:M) предполагает, что в каждый момент времени одному экземпляру информационного объекта A соответствует 0, 1 или более экземпляров объекта B и наоборот (рисунок 5.6).



Рисунок 5.6 – Графическое изображение отношения M:M

Пример 5. Один студент обучается у многих преподавателей, один преподаватель обучает многих студентов, поэтому примером связи M:M может служить связь между информационными объектами *студент* и *преподаватель*:

студент ↔↔↔ преподаватель

Группа связанных между собой таблиц называется *схемой базы данных* (database schema). Информацию о таблицах, их колонках (имена, тип данных, длина поля), первичных и внешних ключах, а также иных объектах базы данных, называются *метаданными* (metadata).

5.2.4 Ссылочная целостность

Одним из правил *ссылочной целостности* (referential integrity) является то, что первичный ключ любой таблицы должен содержать уникальные непустые значения для данной таблицы. Некоторые СУБД могут контролировать уникальность первичных ключей. Если СУБД контролирует уникальность первичных ключей, то при попытке присвоить первичному ключу значение, уже имеющееся в другой записи, СУБД сгенерирует диагностическое сообще-

ние, обычно содержащее словосочетания primary key violation. Это сообщение в дальнейшем может быть передано в приложение, с помощью которого конечный пользователь манипулирует данными.

Если две таблицы связаны соотношением *главная – подчиненная*, внешний ключ подчиненной таблицы должен содержать только те значения, которые имеются среди значений первичного ключа главной таблицы. Если корректность значений внешних ключей не контролируется СУБД, можно говорить о нарушении ссылочной целостности. В этом случае, если мы удалим из таблицы «Студенты» запись, имеющую хотя бы одну связанную с ней подчиненную запись в таблице «Успеваемость», то в таблице «Успеваемость» окажутся записи об оценках по дисциплинам, полученных неизвестно кем.

Если же СУБД контролирует корректность значений внешних ключей, то при попытке присвоить внешнему ключу значение, отсутствующее среди значений первичных ключей *главной* таблицы, либо при удалении или модификации записей *главной* таблицы, приводящих к нарушению ссылочных целостности, СУБД сгенерирует сообщение, о котором говорилось выше.

5.3 ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

5.3.1 Основные этапы проектирования баз данных

Процесс конструирования базы данных (ее проектирования и реализации) будем считать состоящим из последовательности преобразований модели данных одного уровня в модель данных другого уровня. Например, можно рассматривать четыре уровня моделей [2, с. 80]:

- система объектов реального мира;
- информационные структуры, описывающие систему объектов реального мира;
- структуры данных, используемые для представления информационных структур в базе данных и прикладных программах;
- структуры памяти, используемые для хранения структур данных.

Процесс проектирования представляет собой определение метаданных в соответствии с задачами информационной системы, в которой будет использоваться будущая база данных.

Процесс проектирования базы данных включает следующие три этапа:

Первый этап – анализ предметной области или *этап концептуального проектирования*. На этапе концептуального проектирования осуществляется сбор, анализ и редактирование требований к данным.

Второй этап – моделирование построенной информационной системы и проектирование ее отдельных составляющих в форме, соответствующей реальной базе данных или *этап логического проектирования*. В процессе логического проектирования требования к данным преобразуются в структуры используемой СУБД. На этом этапе достаточно ответственным является выбор

СУБД. Это обусловлено тем, что, с одной стороны, число СУБД достаточно велико, а с другой – проектировщику необходимо оценить СУБД по множеству характеристик. Однако основным критерием отбора остается оценка того, насколько эффективно внутренняя модель данных, поддерживаемая системой, способна описать построенную концептуальную схему.

На последнем этапе – *этапе физического проектирования* (тесно связанного с этапом реализации) – решаются вопросы, связанные с производительностью системы, определяются структуры хранения данных и методы доступа.

Процесс проектирования БД не может быть сделан автоматическим, так как для решения многих проблем участие человека является обязательным.

Рассмотрим каждый этап проектирования баз данных более подробно.

5.3.2 Первый этап – анализ предметной области

Первым этапом проектирования БД любого типа является анализ предметной области, который заканчивается построением информационной структуры (концептуальной схемы). На данном этапе анализируются запросы пользователей, выбираются информационные объекты и их характеристики, которые определяют содержание проектируемой БД. На основе проведенного анализа структурируется предметная область. Анализ предметной области не зависит от программной и технической сред, в которых будет реализовываться БД.

Анализ предметной области целесообразно разбить на три фазы:

- 1) анализ концептуальных требований и информационных потребностей;
- 2) выявление информационных объектов и связей между ними;
- 3) построение концептуальной модели предметной области и проектирование концептуальной схемы БД.

Рассмотрим каждую фазу данного этапа проектирования подробно.

Анализ концептуальных требований и информационных потребностей

На этапе анализа концептуальных требований и информационных потребностей необходимо выполнить:

- 1 анализ требований пользователей к базе данных (концептуальных требований);
- 2 выявление имеющихся задач по обработке информации, которая должна быть представлена в базе данных (анализ приложений);
- 3 выявление перспективных задач (перспективных приложений);
- 4 документирование результатов анализа.

Требования пользователей к разрабатываемой БД представляют собой список запросов с указанием их интенсивности и объемов данных. Эти сведения разработчики БД получают в диалоге с ее будущими пользователями. Здесь же выясняются требования к вводу, обновлению и корректировке информации. Требования пользователей уточняются и дополняются при анализе имеющихся и перспективных задач.

Рассмотрим примерный состав вопросника, требований к базе данных

при анализе различных предметных областей.

Пример 1. Пусть предлагается разработать систему вопросов к БД «Сессия студентов вуза»:

1 Цель создания БД?

Ответ. База данных создаётся для информационного обслуживания деканатов факультетов вуза. Одна из основных деятельности деканатов факультетов вуза (на основе положения о факультетах вуза) – учет академической активности студентов, т.е. учет успеваемости студентов. БД должна содержать данные о студентах, дисциплинах, преподавателях вуза, об успеваемости студентов.

2 Сколько студентов учится в вузе?

3 Сколько факультетов и отделений в данном вузе?

4 Как распределены студенты по факультетам отделений и курсам?

5 Сколько дисциплин читается на каждом курсе по каждой специальности?

6 Как часто обновляется информация в базе данных?

7 Сколько преподавателей в вузе?

8 Сколько иногородних студентов живет в общежитии, на частных квартирах?

9 Какая преемственность существует между читаемыми курсами?

10 Сколько лекционных аудиторий и аудиторий для проведения практических занятий, лабораторий?

11 Как информация, представленная в п.п. 1-9, используется в настоящее время (расписание занятий, экзаменов, зачетов и т.д.) и как ее собираются использовать?

12 Сколько раз в день, сколько человек и кто пользуются БД?

Пример 1 (продолжение). Выполним *анализ требований* к БД «Сессия студентов».

Вопрос 1. Для каких типов задач (приложений) проектируется БД?

Ответ. Для четырех типов задач: Задача 1. Информация о студентах. Задача 2. Информация о преподавателях. Задача 3. Информация об успеваемости студентов. Задача 4. Информация о предметах.

Вопрос 2. Какими информационными объектами характеризуются эти задачи?

Ответ. Задача 1 характеризуется информационным объектом: *личные дела студентов*. Задача 2 характеризуется информационным объектом: *личные дела преподавателей*. Задача 3 характеризуется одним информационным объектом – *сессия*. Задача 4 характеризуется одним информационным объектом – *предметы*.

Вопрос 3. Какими информационными атрибутами характеризуются эти информационные объекты?

Ответ. Атрибуты информационного объекта *личные дела студентов*: номер зачетной книжки, ФИО студента, дата рождения, номер группы, образова-

ние, направление подготовки, мед. группа.

Атрибуты информационного объекта *личные дела преподавателей*: табельный номер, ФИО преподавателя, Дата рождения, должность, кафедра, образование, уч. степень, уч. звание.

Атрибуты информационного объекта *сессия*: номер по порядку; номер зач. книжки студента; шифр дисциплины; номер преподавателя; форма отчетности (зачет, экзамен, курсовая), семестр, отметка, дата.

Атрибуты информационного объекта *предметы*: шифр, название, часы (лекции, лаб. работы, семинары), форма отчетности (зачет, экзамен, курсовая), специальность, форма обучения, семестр, программа.

Вопрос 4. Каким текущим запросам должны удовлетворять данные информационные объекты?

Ответ.

- 1) Список студентов группы.
- 2) Список направлений подготовки на данном факультете.
- 3) Список направлений подготовки на данной кафедре (обеспечить не можем).
- 4) ФИО, звание, должность преподавателей кафедры.
- 5) Списочный состав преподавателей, ведущих заданный предмет.
- 6) Список неуспевающих студентов, список студентов на отчисление.
- 7) Список студентов на стипендию.

Вопрос 5. Каким перспективным запросам должны удовлетворять информационные объекты в БД «Сессия студентов»?

Выявление информационных объектов и связей между ними

Вторая фаза анализа предметной области состоит в выборе информационных объектов, задании необходимых свойств для каждого объекта, выявлении связей между объектами, определении ограничений, накладываемых на информационные объекты, типы связей между ними, характеристики информационных объектов.

При выборе информационных объектов следует ответить на следующие вопросы:

- 1 На какие классы можно разбить данные, подлежащие хранению в БД?
- 2 Какое имя можно присвоить каждому классу данных?
- 3 Какие наиболее интересные характеристики (с точки зрения пользователя) каждого класса данных можно выделить?
- 4 Какие имена можно присвоить выбранным наборам характеристик?

В ходе выявления связей между информационными объектами следует ответить на следующие вопросы:

- 1 Какие типы связей между информационными объектами?
- 2 Какое имя можно присвоить каждому типу связей?
- 3 Каковы возможные типы связей, которые могут быть использованы впоследствии?
- 4 Имеют ли смысл какие-нибудь комбинации типов связей?

Далее проектировщик пытается задать ограничения на объекты и их характеристики. Под ограничением целостности обычно понимают логические ограничения, накладываемые на данные. *Ограничение целостности* – это такое свойство, которое проектировщик задает для некоторого информационного объекта или его характеристики, и которое должно сохраняться для каждого их состояния.

При выявлении условий ограничения целостности проектировщик пытается ответить на следующие вопросы:

1 Какова область значений для числовых характеристик?

2 Каковы функциональные зависимости между характеристиками одного информационного объекта?

3 Какой тип отображения соответствует каждому типу связей?

Пример 1 (продолжение). Для БД «Сессия студентов» выберем следующие сущности: *институт, факультет, студент, преподаватель, дисциплина, ведомость*.

Каждую сущность зададим набором атрибутов (ключевые атрибуты подчеркнем):

институт (сокращение, название, подчиненность, адрес, телефон, ФИО ректора).

факультет (код факультета, название, код специальности, декан).

кафедры факультета (код кафедры, название, код факультета, зав.кафедрой).

студент (номер зачетной книжки, ФИО, группа, пол, дата рождения, домашний адрес, телефон).

преподаватель (№ страхового свидетельства, ФИО, дата рождения, домашний адрес, телефон, должность, ученое звание, ученая степень, код кафедры, стаж).

дисциплина (шифр дисциплины, название, число часов, виды занятий, число читаемых семестров, на каких курсах преподается).

ведомость (№ п/п, номер зачетной книжки студента, код дисциплины, семестр, форма сдачи, дата сдачи, отметка, преподаватель).

Определим связи между сущностями.

Имя связи	Связи между сущностями	Имя связи	Связи между сущностями
учится	студент, факультет	работает	преподаватель, факультет
изучает	студент, дисциплина	преподает	преподаватель, дисциплина
Принадлежит	институт, факультет	экзамен	студент, дисциплина преподаватель, студент

Рассмотрим некоторые ограничения на характеристики объектов:

1 Значение атрибута «телефон» (сущность – *институт*) задается целым положительным шестизначным числом, задавать значение будем по маске

__-__-__.

2 Значение атрибута «код факультета» (сущность – *факультет*) лежит в интервале 0-10.

3 Значение атрибута «курс» (сущность – *студент*) лежит в интервале 1-6 и хранится первая цифра номера группы.

4 Значение атрибута «семестр» (сущность – *студент, дисциплина*) лежит в интервале 1–12.

5 Значение атрибута «число часов» (сущность – *дисциплина*) лежит в интервале 1–300.

6 Одному студенту может быть приписана только одна группа.

7 Один студент может учиться только на одном факультете.

8 Один студент в семестре сдает от 3 до 10 дисциплин.

9 Один студент изучает в семестре от 6 до 12 дисциплин.

10 Одному преподавателю приписывается только одна кафедра.

11 Один студент может пересдавать одну дисциплину не более трех раз.

Ключи: сокращение (названия института), код факультета, номер зачетной книжки, № страхового свидетельства преподавателя, шифр дисциплины, № п/п.

Построение концептуальной модели предметной области

Заключительная фаза анализа предметной области состоит в проектировании ее информационной структуры или концептуальной модели.

Концептуальная модель включает описания объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области (ПО) и выявляемых в результате анализа данных. Концептуальная модель применяется для структурирования предметной области с учетом информационных интересов пользователей системы. Она дает возможность систематизировать информационное содержание предметной области, позволяет как бы «подняться вверх» над ПО и увидеть ее отдельные элементы. При этом уровень детализации зависит от выбранной модели. Концептуальная модель является представлением точки зрения пользователя на предметную область и не зависит ни от программного обеспечения СУБД, ни от технических решений. Концептуальная модель должна быть стабильной. Могут меняться прикладные программы, обрабатывающие данные, может меняться организация их физического хранения, концептуальная модель остается неизменной или увеличивается с целью включения дополнительных данных.

Одной из распространенных моделей концептуальной схемы является модель «сущность-связь». Остановимся на наиболее известной модели данного типа, названной по фамилии автора, – модели П. Чена, или ER-модели. Основными конструкциями данной модели являются сущности и связи.

Под *сущностью* понимают основное содержание объекта ПО, о котором собирают информацию. В качестве сущности могут выступать место, вещь, личность, явление. *Экземпляр* сущности – конкретный объект. Например: сущность (объект) – студент, экземпляр сущности – Иванов А. В.; сущность (объект) – институт, экземпляр сущности – КГУ.

Сущность принято определять *атрибутами* – поименованными характе-

ристиками. Например: сущность – студент, атрибуты – ФИО, год рождения, адрес, номер группы и т.д.

Чтобы задать атрибут в модели, ему надо присвоить имя и определить область допустимых значений. Одно из назначений атрибута – идентифицировать сущность.

При построении модели «сущность-связь» используют графические диаграммы (рисунок 5.7). При этом обозначают: сущности – прямоугольниками, атрибуты – овалами, связи – ромбами.

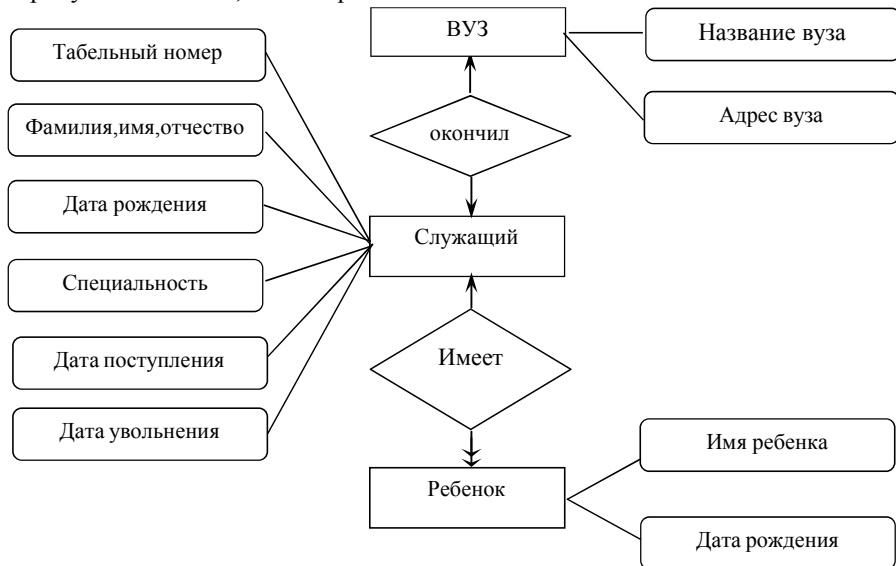


Рисунок 5.7 – Пример модели «сущность – связь»

На практике приходится строить несколько вариантов моделей, из которых выбирается одна, наиболее полно отображающая предметную область.

Пример 1 (продолжение). Спроектировать БД «Сессия студентов». База данных должна выдавать оперативную информацию об успеваемости студентов на факультетах в семестре. Результатами сессии будем считать экзамены и зачеты. По сути дела в БД исходя из формулировки задания можно выделить лишь одно приложение. Речь идет об успеваемости студентов разных факультетов по тем или иным дисциплинам. Более конкретно речь идет о выдаче справок по результатам сессии каждого студента, учебной группы, курса, факультета, а также об автоматизированном составлении ведомости.

Выше мы рассмотрели примерный список вопросов, провели анализ концептуальных требований к БД. Выбрали сущности, указали атрибуты сущностей, установили связи и задали ограничения на характеристики объектов. После выбора сущностей, задания атрибутов и анализа связей можно перейти к проектированию информационной, или концептуальной, схемы БД. Концеп-

туальная схема БД «Сессия студентов» представлена на рисунке 5.8 (атрибуты сущностей на диаграмме не показаны). На рисунке прямоугольниками показаны выявленные сущности, ромбами – связи между сущностями.

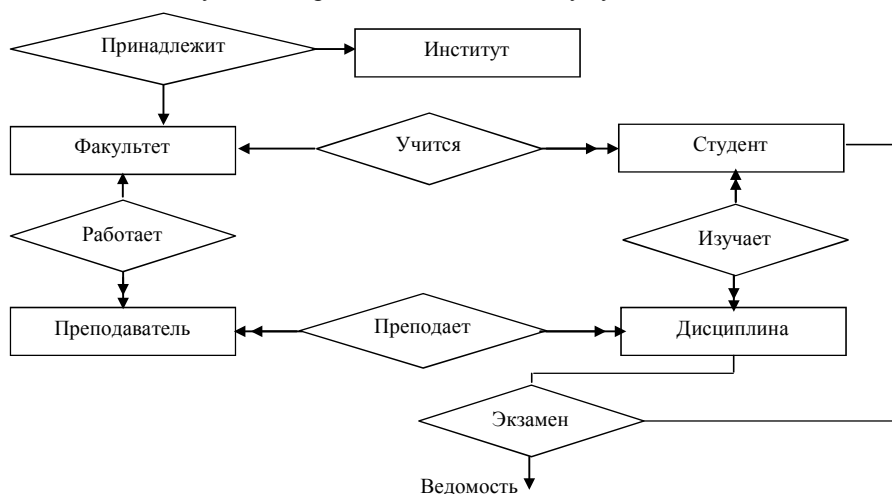


Рисунок 5.8 – Концептуальная схема БД «Сессия студентов»

5.3.3 Второй этап – этап логического проектирования

Логическое проектирование представляет собой необходимый этап при создании БД. Основной задачей логического проектирования является разработка логической схемы, ориентированной на выбранную систему управления базами данных (СУБД). Этап логического проектирования в отличие от концептуального проектирования полностью ориентирован на инструментальные средства компьютера. По сути дела логическое проектирование является моделированием всей информационной системы и ее отдельных составляющих в форме, соответствующей реальной СУБД.

Процесс логического проектирования состоит из следующих этапов:

- 1 выбор конкретной СУБД;
- 2 отображение концептуальной схемы на логическую схему;
- 3 выбор ключей;
- 4 нормализация базы данных;
- 5 описание языка запросов.

Выбор СУБД

Рассмотрим процедуру выбора конкретной СУБД более подробно. Выбор СУБД для практической реализации проекта БД представляется достаточно ответственным. Это обусловлено тем, что, с одной стороны, число СУБД достаточно велико, а с другой – необходимо провести оценку СУБД по большому числу характеристик и выбрать именно ту систему, которая в полной мере сможет удовлетворить требования пользователей и разработчиков по обеспечению

работы с данными. От того, насколько правильно выбрана СУБД, будет зависеть эффективность обработки и использования информации в БД.

Одним из основных критериев выбора СУБД является оценка того, насколько эффективно внутренняя модель данных, поддерживаемая системой, способна описать концептуальную схему. Системы управления базами данных, ориентированные на персональные компьютеры, как правило, поддерживают реляционную или сетевую модель данных. Подавляющее большинство современных СУБД – реляционные. Если выбрана реляционная система, то концептуальную схему БД предстоит отображать на реляционную.

По сути дела выбранная модель данных (реляционная, сетевая или иерархическая) представляет средства для описания структуры данных. Процедуры выполняются на языке описания данных (ЯОД), который входит в ядро СУБД.

Второй составной частью СУБД является язык манипулирования данными (ЯМД), который используется при работе различных приложений с БД. Как правило, ЯМД встраивается в язык программирования. Языки манипулирования данными могут обладать разными возможностями: ЯМД низкого уровня обычно бывают процедурные, высокого уровня – декларативные. Использование процедурных языков требует определенной подготовки, декларативные языки более пригодны для непрофессионального пользователя. Поэтому выбор СУБД, имеющей определенный ЯМД, весьма существен для неподготовленного пользователя. Кроме ядра в СУБД входят сервисные программы и средства для решения прикладных задач. СУБД, ориентированные на персональные компьютеры, отличаются более простой структурой, чем СУБД для мощных ЭВМ.

Отображение информационной схемы на логическую схему БД

Для отображения информационной схемы, полученной на этапе концептуального проектирования, на логическую схему БД необходимо установить:

- 1) размер БД (число сущностей и атрибутов);
- 2) частоту обращения к БД (число выполнения приложений в единицу времени);
- 3) правила построения логических схем БД для имеющихся в распоряжении проектировщика СУБД;
- 4) перечень приложений с указанием используемых данных на БД;
- 5) конфигурацию, быстродействие и объемы памяти, имеющиеся в распоряжении проектировщика технических средств. Отображение информационной схемы на логическую схему БД – это создание из СУБД (независимой структуры БД СУБД) ориентированной логической схемы.

Как уже отмечалось, существующие СУБД делятся по типам моделей данных на реляционные, иерархические, сетевые и комбинированные. Рассмотрим отображение информационной схемы на реляционную модель данных.

Отображение на реляционную модель данных

При отображении информационной схемы каждый прямоугольник схемы для моделей ER в виде ER-диаграммы отображается в таблицу, которая явля-

ется одним отношением. При этом следует учитывать ограничения на размеры таблиц, которые накладывает конкретная СУБД.

Пример 1 (продолжение). Отообразим концептуальную схему базы данных «Сессия студентов» (рисунок 5.8), изображенную в виде ER-диаграммы, на реляционную БД, поддерживаемую СУБД Microsoft Access. Каждая сущность описывается в виде отношения. Первичные ключи помечены #.

Институт

сокращение	название	подчиненность	адрес	телефон	ФИО ректора
#				Текстовое, FK	
Текстовое, 5 символов	Текстовое, 40 символов	Текстовое, 10 символов	Текстовое, 30 символов	8 символов, маска: _-_-_-	Числовое, длинное целое.

На одну запись необходимо $6+41+11+31+9+4 = 102$ байт. На хранение записей в таблице потребуется 102 байт.

Факультет

код факультета	название	код специальности	декан
#			FK
Числовое, длинное целое	Текстовое, 20 символов	Числовое, длинное целое	Числовое, длинное целое.

На одну запись необходимо $4+21+4+4 = 33$ байт. Всего факультетов 11, т.е. на хранение всех записей в таблице потребуется $33 \times 11 = 333$ байт.

Кафедры факультета

код кафедры	название	код факультета	зав. кафедрой
#		FK	FK
Числовое, длинное целое	Текстовое, 20 символов	Числовое, длинное целое	Числовое, длинное целое

На одну запись необходимо $4+21+4+4 = 33$ байт. Всего кафедр 100, т.е. на хранение всех записей в таблице потребуется $33 \times 100 = 3300$ байт.

Дисциплины

Шифр дисциплины	Название	Программа	число часов	виды занятий	Семестр	курс
#						
Числовое, длинное целое	Текстовое, 20 символов	Мето	Числовое, целое, от 1 до 300	Текстовое, 2 символов, список: л, лб, с, п.	Числовое, байт, от 1 до 12	Числовое, байт, от 1 до 5

На одну запись необходимо $4+21+4+3+3+1+1 = 37$ байт. Всего дисциплин 1000, т.е. на хранение всех записей в таблице потребуется $39 \times 1000 = 39000$ байт.

Студенты

№ зач. книжки	Фамилия	Имя	Отчество	Дата рождения
#				
Числовое, длинное целое	Текстовое, 15 символов	Текстовое, 10 символов	Текстовое, 15 символов	Дата/время

<i>Номер группы</i>	<i>Домашний адрес</i>	<i>Домашний телефон</i>
Числовое, целое, от 1000 до 7000	Текстовое, 30 символов	Текстовое, 8 символов, маска: _ _ - _ - _

На одну запись необходимо $4+16+11+11+8+2+31+9=97$ байт. Всего студентов 2000, т.е. на хранение всех записей в таблице потребуется $97 \times 2000 = 184000$ байт.

Преподаватели

<i>№ страх. свидетельства</i>	<i>ФИО</i>	<i>Дата рождения</i>	<i>Домашний адрес</i>	<i>Дом. телефон</i>	<i>Ученое звание</i>
#					Текстовое, 8 символов, список: доцент, профессор, академик
Числовое, длинное целое	Текстовое, 40 символов	Дата/время	Текстовое, 30 символов	Текстовое, 8 символов.	

<i>Ученая степень</i>	<i>Должность</i>	<i>Код кафедры</i>	<i>Стаж</i>
Текстовое, 9 символов., список: кандидат, доктор	Текстовое, 13 символов, список	FK Числовое, длинное целое.	Числовое, байт, от 0 до 50

На одну запись необходимо $4+41+8+31+9+9+10+14+4+1=131$ байт. Всего преподавателей 500, т.е. на хранение всех записей в таблице потребуется 131×500 байт.

Успеваемость студентов

<i>код отметки</i>	<i>Номер зачетной книжки</i>	<i>Шифр дисциплины</i>	<i>Семестр</i>	<i>Форма сдачи</i>
#	FK	FK		Текстовое, 9 символов, список: экзамен, зачет, курсовая, дипломная
Счетчик	Числовое, длинное целое	Числовое, длинное целое	Числовое, байт, от 1 до 12	

<i>Отметка</i>	<i>Дата сдачи</i>	<i>Код преподавателя</i>
Числовое, байт, от 0 до 5	Дата/время	FK Числовое, длинное целое»

На одну запись необходимо $4+4+4+1+10+1+8+4=36$ байт. Всего студентов 2000, каждый студент в семестр примерно сдает 10, учится 12 семестров, т.е. на хранение всех записей в таблице потребуется $36 \times 2000 \times 10 \times 12$ байт.

5.3.4 Третий этап – этап физического проектирования

Задачами физического проектирования являются:

- 1 Определение структуры физической записи.
- 2 Распределение хранимых записей во внешней памяти ЭВМ.
- 3 Выбор метода доступа к хранимым данным.

Эти три задачи решаются при ограничениях на время поиска (не более чем), на цену аппаратных средств (не более чем), на эффективность использо-

вания внешней памяти ЭВМ (не менее чем), на сложность обслуживания БД (обновление, реорганизация и т.д.).

Исходными данными для решения задач физического проектирования являются логическая структура БД и оценки логической структуры.

В современных СУБД выполнение этих трех задач в той или иной степени автоматизировано, поэтому, если читателя не интересуют детали методов решения этих задач, он может пока пропустить данный раздел и перейти к описанию разработки БД средствами конкретных пакетов прикладных программ.

5.4 Контрольные вопросы

1 Данные и информация. Понятия «информация», «данные». Понятие «данные» в концепции баз данных.

2 Модели данных (МД). Понятие о модели данных. Принцип описания МД. Классификация моделей данных

3 Сетевые модели данных. Взаимосвязи в модели. Примеры.

4 Иерархические модели данных. Взаимосвязи в модели. Примеры.

5 Реляционные модели данных. Описание реляционной модели данных. Первичный ключ, внешний ключ, целостность объектов, ссылочная целостная. Виды связей. Взаимосвязи в модели. Примеры.

6 Типы моделей данных: реляционные, иерархические, сетевые. Сравнительная характеристика указанных моделей.

7 Понятие базы данных. Классификация баз данных. Документальные БД. Фактографические БД. Отличия.

8 Типы баз данных в зависимости от типа используемой модели данных. Основные структуры. Отношения между объектами.

9 Сравнительная характеристика основных типов (иерархическая, реляционная, сетевая) баз данных.

10 Функции и содержимое компонентов обработки БД. Системы управления базами данных. Основные функции СУБД. Классификация СУБД. Обзор возможностей и особенностей различных СУБД. Сравнение СУБД.

11 Банк данных (БнД). Состав банка данных. Отличительные особенности банков данных. Состав и структура банка данных. Назначение основных компонентов банка данных. Классификация БнД.

12 Теория реляционных баз данных. Основные требования к реляционной таблице. 12 правил Е.Ф. Кодда. Метаданные.

13 Теория реляционных баз данных. Понятие таблицы (отношения), поля, записи, домена, ключа (первичного, составного первичного и внешнего). Концептуальная схема БД.

14 Теория реляционных баз данных. Понятие главной и дочерней таблиц. Первичные и внешние ключи атрибутов данных. Виды отношений между таблицами.

15 Проектирование баз данных. Полный цикл проектирования базы данных.

16 Этап анализа предметной области (ПО) при проектировании БД. Кон-

цептуальная модель ПО. Требования к концептуальной модели. Основные компоненты концептуальной модели.

17 Этап логического проектирования при проектировании БД. Логическая модель предметной области. Алгоритм перехода от базовой ER-модели к схеме реляционной базы данных.

18 Этап физического проектирования при проектировании БД. Физическое описание модели.

Список источников

1 Бургин М., Степенко Г. Информационный поиск и компьютерная грамотность // Информатика и образование. – 1990. – № 1. – С.15-21.

2 Горбатов В. А., Павлов П. Г., Четвериков В. Н. Логическое управление информационными процессами. – Москва : Энергоатомиздат, 1984. – 304 с.

3 Горев А., Макашариков С., Акаян Р. Эффективная работа с СУБД. – Санкт-Петербург : Питер, 1997. – 900 с.

4 Грей П. Логика, алгебра и базы данных / пер. с англ. ; под ред. Г. В. Орловского. – Москва : Машиностроение, 1989. – 360 с.

5 Гусева Т. И., Башин Ю. И. Проектирование баз данных в примерах и задачах. – Москва : Радио и связь, 1992. – 160 с.

6 Дейт К. Введение в системы баз данных / пер. с англ. – Москва : Наука, 1980. – 464 с.

7 Диго М. С. Проектирование и использование баз данных. – Москва : Финансы и статистика, 1995. – 208 с.

8 Дракин В. И., Попов Э. В., Преображенский А. Б. Общение конечных пользователей с системами обработки данных. – Москва : Радио и связь, 1988. – 288 с.

9 Змитрович А. И. Базы данных : учеб. пособие. – Минск : Университетское, 1991. – 271 с.

10 Информатика в понятиях и терминах : книга для учащихся ст. классов ср. шк. / под ред. В. А. Извозчикова. – Москва : Просвещение, 1991. – 208 с.

11 Канатников А. Н., Ткачев С. Б. Программирование в среде Clipper. Версия 5.0 и особенности версии 5.01. – Москва : Финансы и статистика, 1994. – 240 с.

12 Карманный словарь Хатчинсон. Computing & Multimedia. – Москва : Внешсигма, 1996. – 260 с.

13 Компьютерные технологии обработки информации / под ред. С. В. Назарова. – Москва : Финансы и статистика, 1995. – 248 с.

14 Лекции лауреатов премии Тьюринга / под ред. Ю. М. Баяковского. – Москва : Мир, 1993. – 560 с.

15 Леонов А. Г., Эпиктетов М. Г. Базы данных и электронные таблицы // Информатика и образование. – 1996. – № 3. – С.15-25.

16 Лори П. Базы данных для микро-ЭВМ / пер. с англ. Ю. К. Трубина. – Москва : Машиностроение, 1988. – 136 с.

- 17 Мейер Д. Теория реляционных баз данных / пер. с англ. ; под ред. М. Ш. Цаленко. – Москва : Мир, 1987. – 608 с.
- 18 Озкарахан Э. Машины баз данных и управление базами данных. – Москва : Мир, 1989. – 696 с.
- 19 Персональный компьютер для всех: в 4 кн. / под ред. А. Я. Савельева. – Москва : Высшая школа, 1991. Кн. 3. – 160 с.
- 20 Першиков В. И., Савинков В. М. Толковый словарь по информатике : справ. изд. – Москва : Финансы и статистика, 1991. – 543 с.
- 21 Сиколенко В. К вопросу о тестировании СУБД // СУБД. – 1997. – № 5-6. – С. 80-96.
- 22 Симонович С. В., Евсеев Г. А., Алексеев А. Г. Специальная информатика: учеб. пособие. – Москва : АСТ-ПРЕСС : Инфорком-Пресс, 1998. – 480 с.
- 23 Системы управления базами данных для ЕС ЭВМ : справочник / под общ. ред. В. М. Савинкова. – Москва : Финансы и статистика, 1984. – 224 с.
- 24 Системы управления базами данных и знаний: справ. изд. / под ред. А. Н. Наумова. – Москва : Финансы и статистика, 1991. – 352 с.
- 25 Смирнов Н. Н. Программные средства персональных ЭВМ. – Ленинград : Машиностроение, 1990. – 272 с.
- 26 Ульман Дж. Основы систем и базы данных. – Москва : Финансы и статистика, 1983. – 336 с.
- 27 Федоров А., Елманова Н. Введение в базы данных // Компьютер Пресс. – 2000. – март – С. 146 - 149.

6 КОНТРОЛЬ И ЗАЩИТА ИНФОРМАЦИИ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ

6.1 ОСНОВНЫЕ ПОНЯТИЯ ЗАЩИТЫ ИНФОРМАЦИИ

В настоящее время методы передачи, хранения и обработки информации спровоцировали появление таких угроз, как искажение, раскрытие и потери данных, которые передаются или принадлежат различным пользователям. Вследствие этого одним из главных направлений развития информационных технологий стало обеспечение информационной безопасности компьютерных систем и сетей.

Рассмотрим основные понятия, связанные с защитой информации.

В федеральном законе «Об информации, информационных технологиях и о защите информации» даются следующие понятия:

1) информация – сведения (сообщения, данные) независимо от формы их представления;

...

5) обладатель информации – лицо, самостоятельно создавшее информацию либо получившее на основании закона или договора право разрешать или ограничивать доступ к информации, определяемой по каким-либо признакам;

6) доступ к информации – возможность получения информации и ее использования;

7) конфиденциальность информации – обязательное для выполнения лицом, получившим доступ к определенной информации, требование не передавать такую информацию третьим лицам без согласия ее обладателя;

8) предоставление информации – действия, направленные на получение информации определенным кругом лиц или передачу информации определенному кругу лиц;

9) распространение информации – действия, направленные на получение информации неопределенным кругом лиц или передачу информации неопределенному кругу лиц [3].

Под защитой информации понимаются различные меры, направленные на предотвращение утечки информации, несанкционированных, непреднамеренных и преднамеренных воздействий на информацию.

Объектом защиты информации является автоматизированная система обработки данных или компьютерная система. В качестве компьютерной системы рассматривается комплекс аппаратных и программных средств, предназначенных для автоматизированного сбора, обработки, хранения, получения и передачи информации.

Понятие компьютерной системы очень широкое и оно охватывает следующие системы:

- ЭВМ всех классов и назначений;
- вычислительные комплексы и системы;

- сети и телекоммуникации.

Предметом защиты в компьютерной системе является информация. Информация в компьютерной системе материально представляется в виде электромеханических и электронных устройств, а также машинных носителей. Информация попадает в компьютерные системы с помощью систем передачи данных или устройств ввода. В системе информация хранится в запоминающих устройствах различных уровней, обрабатывается (преобразуется) процессорами и выводится из системы с помощью устройств вывода или систем передачи данных. В качестве машинных носителей может быть использована бумага, диски различных типов. Большинство типов машинных носителей информации – съемные, т.е. могут сниматься с устройств и использоваться или храниться отдельно от устройств.

Таким образом, для защиты информации (обеспечения безопасности информации) в компьютерных системах необходима защита машинных носителей и устройств от несанкционированных воздействий на них.

Одним из основных понятий теории защиты информации является понятие «информационная безопасность».

Под информационной безопасностью понимают защищенность информации от незаконного ознакомления, преобразования и уничтожения, а также защищенность информационных ресурсов от воздействий, направленных на нарушение их работоспособности [12]. Защищенными называются компьютерные системы, в которых обеспечивается безопасность информации.

Под эффективностью защиты информации понимается степень соответствия результатов защиты информации согласно поставленной цели.

Защита информации от утечки представляет собой деятельность по предотвращению неконтролируемого распространения информации от ее разглашения, несанкционированного доступа к информации и от получения информации злоумышленниками.

К непреднамеренным воздействиям на информацию относятся:

- ошибки пользователей информацией,
- сбой технических и программных средств информационных систем,
- природные явления,

- иные нецеленаправленные воздействия на изменение информации, связанные с функционированием технических средств, систем или с деятельностью людей и приводящих к искажению, уничтожению, копированию, блокированию доступа к информации, а также к утрате, уничтожению или сбою функционирования носителя информации.

К несанкционированным воздействиям относятся воздействия на информацию с нарушением установленных прав или правил на изменение информации, приводящего к уничтожению, копированию, искажению, блокированию доступа к информации, а также к сбою, уничтожению или утрате функционирования носителя информации.

Защитой информации от непреднамеренного и несанкционированного

воздействий является деятельность по предотвращению перечисленных выше воздействий.

Защита информации от несанкционированного доступа связана с деятельностью по предотвращению получения информации заинтересованным субъектом с нарушением установленных собственником или правовыми документами правил или прав доступа к защищаемой информации. Заинтересованными субъектами, которые осуществляют несанкционированный доступ к информации, могут выступать юридические лица, государство, группа физических лиц, общественная организация или отдельные физические лица.

Система защиты информации состоит из совокупности органов или исполнителей, используемая ими техника защиты информации, а также объекты защиты, организованные и функционирующие по правилам, установленным соответствующими правовыми, нормативными и организационно-распорядительными документами по защите информации.

6.2 УГРОЗЫ БЕЗОПАСНОСТИ ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ

Безопасность информации компьютерных систем рассматривают в виде единства трех составляющих, которые оказывают взаимное влияние друг на друга:

- технические и программные средства;
- информация;
- обслуживающий персонал и пользователи.

Компьютерная система разрабатывается для удовлетворения потребностей пользователей в получении достоверной информации и конфиденциального ее сохранения. Информация является конечным «продуктом потребления» в компьютерной системе и выступает в виде центральной компоненты системы. Безопасность информации на уровне компьютерной системы обеспечивают две другие компоненты системы. Причем эта задача должна решаться путем защиты от внешних и внутренних неразрешенных (несанкционированных) воздействий (угроз) [9].

Под угрозой безопасности информации понимается потенциально возможное событие, процесс или явление, которые могут привести к уничтожению, утрате целостности, конфиденциальности или доступности информации [5].

Все множество потенциальных угроз безопасности информации в компьютерной системе может быть разделено на два класса (рисунком 6.1):

6.2.1 Виды случайных угроз

Если угрозы не относятся к преднамеренным действиям злоумышленников и создаются в случайные моменты времени, то они называются случайными или непреднамеренными.

Класс случайных угроз вызывает наибольшее число потери информации.

При этом может происходить нарушение целостности, доступности и уничтожение информации. Иногда разрушается конфиденциальность информации, и злоумышленники могут воздействовать на информацию.

Рассмотрим случайные угрозы.

Стихийные бедствия и аварии наиболее опасны разрушительными последствиями для компьютерной системы, так как они подвергают компьютерные системы физическому разрушению, информация теряется или доступ к ней становится невозможен.

Сбои и отказы технических средств неминуемы. Сбои и отказы технических средств приводят к:

- искажению и уничтожению программ и данных;
- нарушению работоспособности технических средств;
- нарушению алгоритмов работы устройств, что может привести к нарушению конфиденциальности информации.

Например, к несанкционированной выдаче информации в каналы связи, на печатающие устройства и т.д. могут привести сбои и отказы средств выдачи информации.



Рисунок 6.1 – Угрозы безопасности в компьютерных системах

Ошибки при разработке компьютерной системы, алгоритмические и про-

граммные ошибки могут привести к точно таким же последствиям, как и сбои и отказы технических средств. Более того, эти ошибки могут быть использованы злоумышленниками для воздействия на ресурсы компьютерной системы. Особую опасность представляют ошибки в программных средствах защиты информации и в операционных системах.

Согласно данным Национального Института Стандартов и Технологий США 65% случаев нарушения безопасности информации происходит в результате ошибок пользователей и обслуживающего персонала [9]. Если сотрудники некомпетентны, небрежны или невнимательны при выполнении своих функциональных обязанностей, то это может привести к нарушению целостности и конфиденциальности информации, уничтожению, а также оглашению сведений механизмов защиты.

Рассматривая угрозы информации в компьютерной системе, которые не связаны с преднамеренными действиями, можно сказать, что механизм их реализации изучен на достаточно хорошем уровне, накоплен значительный опыт противодействия этим угрозам. Эффективная система эксплуатации компьютерной системы, включающая обязательное резервирование информации, а также современные технологии разработки программных и технических средств позволяют значительно снизить потери от всех видов случайных угроз.

6.2.2 Виды преднамеренных угроз

Преднамеренно создаваемые угрозы создают второй класс угроз безопасности информации в компьютерной системе.

Данный класс угроз очень активен и постоянно дополняется новыми угрозами. Угрозы этого класса в соответствии с их механизмами реализации и физической сущностью могут быть разделены на следующие группы:

1 Несанкционированный доступ к информации, т.е. это доступ к информации, нарушающий правила разграничения доступа с использованием средств вычислительной техники или автоматизированных систем.

Под правилами разграничения доступа понимается совокупность положений, описывающих права доступа лиц или процессов, являющимися субъектами доступа, к единицам информации, являющимися объектами доступа.

К ресурсам компьютерной системы право доступа устанавливается руководством для каждого сотрудника согласно с его функциональными обязанностями. Процессы происходят в компьютерной системе в интересах определенных лиц, поэтому на них устанавливаются ограничения по доступу к ресурсам.

Выполнение установленных правил разграничения доступа в компьютерную систему осуществляются за счет создания системы разграничения доступа.

Несанкционированный доступ к информации возможен только с использованием программных и аппаратных средств в следующих случаях:

- отказ или сбой в компьютерной системе;
- не создана система разграничения доступа;
- ошибочные действия обслуживающего персонала или пользователей

компьютерных систем;

- ошибки в системе разграничения доступа;
- подмена полномочий.

Злоумышленник, который имеет навыки работы в компьютерной системе, может получить без ограничений доступ к любой информации, если система разграничения доступа отсутствует. Возможны состояния системы, при которых упрощается несанкционированный доступ к информации, в результате сбоев или отказов средств компьютерной системы, а также ошибочных действий пользователей и обслуживающего персонала. Злоумышленник может обнаружить ошибки в системе разграничения доступа и применить их для несанкционированного доступа к информации. Подмена полномочий может быть одним из наиболее вероятных путей или каналов несанкционированного доступа к информации.

2 Традиционный шпионаж и диверсии. Методы и средства шпионажа и диверсий, которые используются для уничтожения или добывания информации на объектах, не имеющих компьютерной системы, могут быть использованы в качестве источников нежелательного воздействия на информационные ресурсы. Они также действенны и результативны в условиях использования компьютерных систем. Чаще всего они используются для получения сведений о системе защиты с целью проникновения в компьютерную систему, а также для хищения и уничтожения информационных ресурсов.

К методам шпионажа и диверсий относятся [9]:

- подслушивание;
- визуальное наблюдение;
- хищение документов и машинных носителей информации;
- хищение программ и атрибутов системы защиты;
- подкуп и шантаж сотрудников;
- сбор и анализ отходов машинных носителей информации;
- поджоги;
- взрывы.

3 Несанкционированная модификация структур. Значительной угрозой безопасности информации в компьютерной системе является несанкционированная модификация программной, алгоритмической и технической структур системы. Данная модификация структур может происходить на любом жизненном цикле компьютерной системы. Несанкционированное изменение структуры компьютерной системы на этапах разработки и изменении получило название «закладка» [9].

При разработке компьютерной системы «закладки» внедряются в специализированные системы, которые предназначены для использования в различных фирмах или государственных учреждениях. «Закладки», которые внедрены на этапе разработки, сложно найти, так как их авторы имеют высокую квалификацию, и из-за сложности современных компьютерных систем.

Программные, алгоритмические и аппаратные «закладки» используются

либо для прямого вредительского воздействия на компьютерную систему, либо для обеспечения неконтролируемого входа в систему. При получении соответствующей команды извне и при наступлении определенных событий в системе осуществляются вредоносные воздействия «закладок» на компьютерные системы.

К таким событиям можно отнести:

- переход на определенный режим работы;
- наступление установленной даты;
- достижение определенной цели и т. д.

4 Электромагнитные излучения и наводки. Процессы передачи и обработки информации техническими средствами компьютерной системы сопровождаются электромагнитными излучениями в окружающей среде и вызыванием электрических сигналов в сигнализации, линиях связи, заземлении и других проводниках. Они получили названия побочных электромагнитных излучений и наводок [9]. Сигналы принимаются, выделяются, усиливаются и могут просматриваться или записываться в запоминающих устройствах с помощью специального оборудования.

Наведенные в проводниках электрические сигналы могут выделяться и фиксироваться при помощи оборудования, которое подключается к этим проводникам на расстоянии в сотни метров от источника сигналов. Для добывания информации злоумышленник может использовать также «просачивание» информационных сигналов в цепи электропитания технических средств компьютерной системы.

Электромагнитные излучения злоумышленниками используются для уничтожения информации, а не только для ее получения. Вывести из строя электронные блоки компьютерной системы могут мощные электромагнитные и сверхвысокочастотные излучения.

5 Вредительские программы – это специальные программы, которые являются одним из основных источников угроз безопасности информации в компьютерной системе.

В зависимости от механизма действия вредительские программы делятся на следующие классы [9]:

- «Червями» называются программы, которые выполняются каждый раз при загрузке системы, обладают способностью перемещаться в вычислительной системе или сети и самовоспроизводить копии. Лавинообразное размножение программ приводит к перегрузке каналов связи, памяти и, в конечном итоге, к блокировке системы;

- «Троянские кони» – это программы, полученные путем явного изменения или добавления команд в пользовательские программы. При последующем выполнении пользовательских программ наряду с заданными функциями выполняются несанкционированные, измененные или какие-то новые функции;

- «Компьютерные вирусы» – это небольшие программы, которые после внедрения в ЭВМ самостоятельно распространяются путем создания своих ко-

пий, а при выполнении определенных условий оказывают негативное воздействие на компьютерную систему. Поскольку вирусам присущи свойства всех классов вредительских программ, то любые вредительские программы часто называют вирусами.

Помимо этих угроз безопасности существует также угроза утечки информации, которая становится с каждым годом все более серьезной проблемой безопасности.

На четыре основных типа утечек приходится подавляющее большинство (84%) инцидентов, причем половина этой доли (40%) приходится на самую популярную угрозу – кражу носителей. 15% составляет инсайд. К данной категории относятся инциденты, причиной которых стали действия сотрудников, имевших легальный доступ к информации. Например, сотрудник не имел права доступа к сведениям, но сумел обойти системы безопасности. Или инсайдер имел доступ к информации и вынес ее за пределы организации. На хакерскую атаку также приходится 15% угроз. В эту обширную группу инцидентов попадают все утечки, которые произошли вследствие внешнего вторжения. Не слишком высокая доля хакерских вторжений объясняется тем, что сами вторжения стали незаметнее. 14% составила Web-утечка. В данную категорию попадают все утечки, связанные с публикацией конфиденциальных сведений в общедоступных местах, например, в Глобальных сетях. 9% - это бумажная утечка. По определению бумажной утечкой является любая утечка, которая произошла в результате печати конфиденциальных сведений на бумажных носителях. 7% составляют другие возможные угрозы. В данную категорию попадают инциденты, точную причину которых установить не удалось, а также утечки, о которых стало известно постфактум, после использования персональных сведений в незаконных целях [11].

Также в настоящее время активно развивается фишинг-технология Интернет-мошенничества, заключающаяся в краже личных конфиденциальных данных, таких как:

- пароли доступа;
- номера кредитных карт;
- банковских счетов;
- другой персональной информации.

Фишинг (от англ. Fishing – рыбалка) расшифровывается как выуживание пароля и использует не технические недостатки компьютерной системы, а легковерность пользователей Интернета. Злоумышленник закидывает в Интернет приманку и «вылавливает всех рыбок» – пользователей, которые на это клюют [6].

Независимо от особенностей конкретных видов угроз, информационная безопасность должна сохранять доступность, целостность, конфиденциальность. Угрозы нарушения доступности, целостности и конфиденциальности являются первоначальными. Угроза недоступности информации может возникнуть тогда, когда в результате преднамеренных действий других пользователей

или злоумышленников организовывается блокировка доступа к некоторому ресурсу компьютерной системы. Нарушение целостности включает в свой состав любое умышленное изменение информации, которое хранится в компьютерной системе или передается из одной системы в другую. Нарушение конфиденциальности иногда приводит к ситуации, при которой информация становится известной тому, кто не располагает полномочиями доступа к ней.

6.3 КОМПЛЕКСНЫЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Защита информации в компьютерных системах предоставляется формированием комплексной системы защиты. Комплексная система защиты включает:

- правовые методы защиты;
- организационные методы защиты;
- криптографические методы защиты;
- стеганографические методы защиты.

6.3.1 Правовые методы защиты

К правовым мерам и средствам защиты относятся:

- действующие в стране законы;
- нормативные акты, регламентирующие правила обращения с информацией и ответственность за их нарушение.

Согласно Конституции РФ [1]:

- «каждый имеет право на неприкосновенность частной жизни, личную и семейную тайну, на тайну переписки, телефонных переговоров, почтовых, телеграфных и иных сообщений» (статья 23);
- «сбор, хранение, использование и распространение информации о частной жизни лица без его согласия не допускаются» (статья 24);
- «каждый имеет право свободно искать, получать, передавать, производить и распространять информацию любым законным способом, перечень сведений, составляющих государственную тайну, определяется федеральным законом» (статья 29);
- «каждый имеет право на достоверную информацию о состоянии окружающей среды» (статья 42).

Основополагающим законодательным актом в России, регулирующим отношения в информационной сфере, в том числе связанные с защитой информации, является Федеральный закон «Об информации, информационных технологиях и о защите информации», принятый 27 июля 2006 года (№149-ФЗ).

Предметом регулирования данного Закона являются общественные отношения, возникающие в трех взаимосвязанных направлениях [3]:

- создание и использование информационных технологий и средств их обеспечения;
- формирование и использование информационных ресурсов;

- защита информации, прав субъектов, участвующих в информационных процессах и информатизации.

В Законе даны основные понятия в информационной сфере.

В Законе также освещены вопросы, связанные с:

порядком обращения с персональными данными;

- сертификацией информационных систем, технологий, средств их обеспечения;

- лицензированием деятельности по формированию и использованию информационных ресурсов.

Одно из главных мест в системе законодательства любого государства занимает институт государственной тайны. Причина этого – величина ущерба, который может быть нанесен государству, если будут разглашены сведения, составляющие государственную тайну.

Закон «О государственной тайне» относится к специальным законодательным актам, регулирующим отношения, которые возникают в связи с отнесением сведений к государственной тайне, их защитой и рассекречиванием.

Согласно Закону, государственная тайна – защищаемые государством сведения в области его военной, внешнеполитической, экономической, разведывательной, контрразведывательной и оперативно-розыскной деятельности, распространение которых может нанести ущерб безопасности Российской Федерации [2].

К средствам защиты информации данным Законом отнесены:

- технические средства;

- криптографические средства;

- программные средства;

- средства, предназначенные для защиты сведений, составляющих государственную тайну;

- средства, в которых они реализованы;

- средства контроля эффективности защиты информации.

Указом от 6 марта 1997 года № 188 Президент Российской Федерации утвердил Перечень сведений конфиденциального характера, в котором выделены шесть основных категорий информации [4]:

1 персональные данные;

2 тайна следствия и судопроизводства;

3 служебная тайна;

4 профессиональные виды тайн (врачебная, нотариальная, адвокатская и т.д.);

5 коммерческая тайна;

6 сведения о сущности изобретения, полезной модели или промышленного образца до официальной публикации информации о них.

Органы государственной власти РФ выполняют основную роль в создании правовых механизмов защиты информации.

Поручителем Конституции РФ, прав и свобод, в том числе информаци-

онных, человека и гражданина является Президент РФ. Он отвечает за работу федеральных органов исполнительной власти, которые управляют вопросами безопасности, издает указы и распоряжения по вопросам информационной безопасности и защиты информации.

Федеральное Собрание формирует законодательную базу в области защиты информации. Федеральное Собрание – парламент Российской Федерации. Он состоит из двух палат – Совета Федерации и Государственной Думы, является законодательным органом Российской Федерации. В структуре Государственной Думы имеется Комитет по информационной политике. Комитет организует законотворческую деятельность в информационной сфере. Им разработана Концепция государственной информационной политики, содержащая раздел, посвященный информационному законодательству. Также в подготовке законопроектов, которые направлены на улучшение законодательства в области защиты информации, участвуют и другие комитеты Государственной Думы.

Совет Безопасности Российской Федерации формируется Президентом Российской Федерации. Он является одним из органов, связанных с нормативным правовым регулированием в области защиты информации. Межведомственная комиссия Совета Безопасности Российской Федерации по информационной безопасности была создана указом Президента Российской Федерации от 19 сентября 1997 года № 1037 в целях реализации задач в области обеспечения информационной безопасности Российской Федерации, возложенных на Совет Безопасности Российской Федерации. Одной из задач этой комиссии является подготовка предложений по нормативному правовому регулированию вопросов информационной безопасности и защиты информации. Аппаратом Совета Безопасности в соответствии с Концепцией национальной безопасности Российской Федерации был подготовлен проект Доктрины информационной безопасности Российской Федерации.

Межведомственная комиссия по защите государственной тайны была образована Указом Президента Российской Федерации от 8 ноября 1995 года № 1108 в целях осуществления единой государственной политики в области засекречивания информации. Она также следит за деятельностью органов государственной власти по защите государственной тайны в интересах разработки и выполнения государственных программ и нормативных документов. Комиссия играет особую роль в процессе формирования нормативной правовой базы в области защиты информации.

Федеральные органы государственной власти, органы государственной власти субъектов Российской Федерации, органы местного самоуправления, предприятия, учреждения, организации, должностные лица и граждане обязаны исполнять решения Межведомственной комиссии по защите государственной тайны, которые приняты в соответствии с ее полномочиями.

Центральному аппарату Государственной технической комиссии при Президенте Российской Федерации (Гостехкомиссии России) поручено организа-

ционно-техническое обеспечение деятельности Межведомственной комиссии. Гостехкомиссия России решает проблемы защиты информации в Российской Федерации и является одним из основных органов власти по этим проблемам. Она является федеральным органом исполнительной власти, осуществляющим межотраслевую координацию и функциональное регулирование деятельности по обеспечению некриптографическими методами защиты информации, содержащей сведения, составляющие государственную или служебную тайну. Эти методы предохраняют от ее утечки по техническим каналам, от несанкционированного доступа к ней, от специальных воздействий на информацию в целях ее уничтожения, искажения и блокирования.

Гостехкомиссия разработала проект Концепции совершенствования правового обеспечения информационной безопасности Российской Федерации. В нем были описаны основные направления совершенствования законодательства в области информационной безопасности. Он был разработан рабочей комиссией при аппарате Совета Безопасности Российской Федерации.

Совершенствование законодательства субъектов Российской Федерации направлено на формирование в рамках единой системы обеспечения информационной безопасности Российской Федерации региональных систем обеспечения информационной безопасности субъектов Российской Федерации.

Сегодня существует необходимость в совершенствовании нормативной правовой базы в области информационной безопасности и защиты информации, несмотря на то, что в Российской Федерации за довольно короткое время сформировалась достаточно обширная нормативно-правовая база [7].

6.3.2 Организационные методы защиты

Организационные мероприятия по защите включают в себя совокупность действий по подбору и проверке персонала, участвующего в подготовке и эксплуатации программ и информации, строгое регламентирование процесса разработки и функционирования компьютерных систем [10].

Данные методы защиты состоят из мер, мероприятий и действий, которые должны выполнять должностные лица в процессе создания и эксплуатации компьютерных систем для обеспечения заданного уровня безопасности информации.

На предприятиях, в ведомствах, министерствах согласно законам и нормативным актам для защиты информации создаются специальные службы безопасности, которые подчиняются напрямую руководству учреждения. Руководители служб организуют создание и функционирование системы защиты информации.

На организационном уровне решаются следующие задачи обеспечения безопасности информации в компьютерных системах:

- организуется работа по разработке системы защиты информации;
- ограничивается доступ на объекты и к ресурсам компьютерных систем;

- разграничивается доступ к ресурсам компьютерных систем;
- планируются мероприятия, направленные на обеспечение защиты информации;
- разрабатывается документация;
- обучается и воспитывается обслуживающий персонал и пользователи;
- сертифицируются средства защиты информации;
- лицензируется деятельность по защите информации;
- аттестуются объекты защиты информации;
- совершенствуются системы защиты информации;
- оценивается эффективность функционирования системы защиты информации;
- контролируется выполнение установленных правил работы в компьютерных системах.

Организационные методы представляют из себя стержень комплексной системы защиты информации в компьютерных системах. Данные методы объединяют на правовой основе технические, программные и криптографические средства защиты информации в единую комплексную систему.

6.3.3 Криптографические методы защиты

Суть криптографической защиты заключается в преобразовании информации к неясному виду с помощью специальных алгоритмов либо аппаратных средств и соответствующих кодовых ключей.

Наука о создании и анализе систем безопасности связи называется криптологией.

Криптология состоит из двух частей: криптографии и криптоанализа.

Криптографией называется наука о методах обеспечения секретности и подлинности (идентичности) данных при их передаче по линиям связи или хранении.

Криптоанализом называется наука о методах раскрытия или подделки данных.

Из определений видно, что криптография и криптоанализ нацелены на решение взаимно обратных задач.

Преобразование исходной информации, в результате которого она становится недоступной для ознакомления и использования лицами, которые не имеют на это полномочий, называется криптографической защитой информации.

Известны различные подходы к классификации методов криптографического преобразования информации. Дадим классификацию методов криптографического преобразования информации по виду воздействия на исходную информацию. По данному признаку их можно разделить на три группы (рисунок 6.2)



Рисунок 6.2 – Классификация методов криптографического преобразования информации

Рассмотрим каждый из этих методов.

Шифрованием называется способ преобразования исходной информации в шифртекст и обратно.

Шифрование используется для того, чтобы обычный текст, называемый открытым текстом, преобразовать в секретный, называемый зашифрованным текстом. Шифрование обеспечивает неизменность данных во время передачи сообщения и дает возможность проверить личность отправителя.

При использовании процессов шифрования проводятся обратимые математические, логические, комбинаторные и другие преобразования исходной информации. В результате этих преобразований зашифрованная информация представляет собой неупорядоченный набор букв, цифр, других символов и двоичных кодов.

Для шифрования информации применяется алгоритм преобразования и ключ. Обычно алгоритм шифрования для определенного метода не изменяется. Исходными данными для алгоритма шифрования служат информация, которую нужно зашифровать, и ключ шифрования. Ключ имеет управляющую информацию, которая делает выбор преобразования на определенных шагах алгоритма и величины операндов, которые используются для реализации алгоритма шифрования.

Методы шифрования в зависимости от структуры используемых ключей делят на:

- симметричное шифрование: сторонние лица могут знать алгоритм шифрования, но им неизвестен ключ, который является одинаковым для отправителя и получателя сообщения;
- асимметричное шифрование: сторонние лица могут знать алгоритм шифрования, и, возможно, открытый ключ, но им неизвестен закрытый ключ, известный только получателю.

Методы шифрования можно разделить на три основных вида:

- симметричная криптография;
- асимметричная криптография;
- хеширование.

Каждый из этих методов шифрования имеет свои достоинства и недостатки. Например, хеширование является стойким к вмешательству, но он не очень гибок как другие методы.

Рассмотрим эти виды.

Симметричное шифрование также называют шифрованием с секретным ключом. Термин «секретный ключ» используется потому, что при шифровании и расшифровании данных используется ключ, который должен остаться засекреченным. Любой обладатель данного ключа может узнать закодированное сообщение. Отправитель шифрует информацию, используя ключ, а получатель использует тот же ключ для расшифрования данных.

Эти методы шифрования применяются для шифрования потока или блоков данных. Это зависит от объема данных, которые нужно зашифровать или расшифровать за один раз. В режиме потока шифруется каждая отдельная единица информации. А при блочном методе шифруются отдельные блоки информации. Самые известные и применяемые методы симметричного кодирования: DES, AES, IDEA.

Асимметричные методы шифрования или методы шифрования информации с открытым ключом, более надежны, чем симметричные методы шифрования. Это тип шифрования применяет два ключа: секретный и открытый ключ. Секретный ключ, используется для расшифровки данных. Он имеется только у получателя. Открытый ключ используется для шифрования информации и передается в открытом канале. Самыми используемыми асимметричными методами шифрования являются методы RSA и Diffie-Hellman.

При применении шифрования по методу хеширования для сообщения или набора данных создается уникальная подпись фиксированной длины. Специальным алгоритмом или хеш-функцией создаётся хеш. Он используется для сравнения данных. Для каждого набора данных или сообщения хеш является уникальным, поэтому небольшое изменение данных приведет к сильному отличию хеша. Это будет показывать отличие двух одинаковых данных.

Метод хеширования обладает, в отличие от других методов шифрования, тем, что после шифрования хеш не может быть расшифрован или изменен. Т.е. если злоумышленник получит хеш код, он не сможет его декодировать и получить исходное сообщение.

С помощью процесса кодирования информации исходная информация заменяется кодами. Сочетания букв, цифр, букв и цифр могут применяться в качестве кодов. Для того, чтобы провести кодирование и его обратное преобразование используются специальные таблицы или словари. Кодирование информации рационально применять в системах с ограниченным набором смысловых конструкций.

Для того чтобы кодировать информацию, нужно хранить и распространять кодировочные таблицы. Также их нужно часто менять, чтобы избежать раскрытия кодов статистическими методами обработки перехваченных сообщений. Это является недостатком использования кодирования для создания конфиденциальной информации.

Сжатие информации относится к методам криптографического преобразования информации, но с определенными ограничениями. Сокращение объема информации является основной целью ее сжатия. Без обратного преобразова-

ния СЖАТАЯ информация не может быть прочитана или использована. Средства сжатия и обратного преобразования являются доступными. Методы сжатия не считаются надежными средствами криптографического преобразования информации. Они могут быть относительно легко раскрыты статистическими методами обработки, даже если держать в секрете алгоритмы сжатия. Поэтому, если сжатие используется для криптографического преобразования информации, то сжатые файлы секретной информации подвергаются последующему шифрованию. Причем, для сокращения времени преобразования информации, желательно совмещать процесс ее сжатия и шифрования.

6.3.4 Стеганографические методы защиты

Стеганография, с греческого «тайнопись», – наука о скрытой передаче данных.

Методы стеганографии, в противоположность от других методов криптографического преобразования информации, дают возможность скрыть не только смысл передачи или хранения информации, но и сам факт передачи или хранения секретной информации. Стеганография изучает методы передачи скрытой информации через ненадежные каналы связи. Маскировка секретной информации среди открытых файлов лежит в основе всех методов стеганографии.

Компьютерная стеганография включает в себя следующие направления: встраивание информации с целью ее скрытой передачи; встраивание цифровых водяных знаков; встраивание идентификационных номеров; встраивание заголовков [8].

Стеганографию разделяют на три раздела:

- классическую стеганографию. Он включает в себя все «некомпьютерные методы»;
- компьютерную стеганографию. Включает в себя направления классической стеганографии, которые основаны на особенностях компьютера и использовании специальных свойств компьютерных форматов данных;
- цифровая стеганография. Включает направления классической стеганографии, основанные на сокрытии или внедрении дополнительной информации в цифровые объекты, вызывая при этом некоторые искажения этих объектов. В этих направлениях используется избыточность аудио- и визуальной информации.

Рассмотрим, как можно применить стеганографические методы защиты информации для передачи скрытой информации.

Графическую информацию можно представить в числовом виде. Наи-

меньший элемент изображения в графических объектах можно закодировать одним байтом. В младшие разряды определенных байтов изображения в соответствии с алгоритмом криптографического преобразования помещаются биты скрытого файла. Если правильно подбираются алгоритм преобразования и изображение, на фоне которого помещается скрытый файл, то человеческому глазу практически невозможно отличить полученное изображение от исходного. Такую скрытую информацию очень сложно найти даже при помощи специальных программ. В качестве изображения лучше всего для внедрения скрытой информации подходят снимки местности: фотоснимки со спутников, самолетов и т. п.

При помощи методов стеганографии можно замаскировать изображение, текст, речь, цифровую подпись, зашифрованное сообщение. Одновременное использование методов стеганографии и шифрования в несколько раз повышает сложность решения задачи обнаружения и раскрытия конфиденциальной информации.

6.4 Контрольные вопросы

- 1 Дайте понятия информации и обладателя информации. Что такое доступ к информации, конфиденциальность информации, предоставление информации?
- 2 Что понимают под информационной безопасностью?
- 3 Перечислите виды случайных угроз безопасности информации.
- 4 Перечислите виды преднамеренных угроз безопасности информации.
- 5 Перечислите основные правовые документы, регулирующие вопросы защиты информации в компьютерных системах
- 6 Перечислите задачи, решаемые на организационном уровне.
- 7 В чем состоит суть криптографических методов защиты информации?
- 8 Дайте понятия криптологии, криптографии и криптоанализа.
- 9 Дайте классификацию методов криптографического преобразования информации по виду воздействия на исходную информацию. Охарактеризуйте их.
- 10 Перечислите разделы стеганографии. Охарактеризуйте их.

Список источников

- 1 Конституция Российской Федерации : текст с посл. изменениями на 2016 г. – Москва : Эксмо, 2016. – 32 с.
- 2 О государственной тайне : федер. закон от 21 июля 1993 г., № 5485-1-ФЗ (ред. от 08.03.2015) [Электронный ресурс] // КонсультантПлюс : справ. правовая система.
- 3 Об информации, информационных технологиях и о защите информации : федер. закон от 27 июля 2006 г., № 149-ФЗ [Электронный ресурс] // КонсультантПлюс : справ. правовая система.
- 4 Об утверждении перечня сведений конфиденциального характера :

Указ Президента РФ от 6 марта 1997 г. № 188 (с изм. и доп.) [Электронный ресурс] // КонсультантПлюс : справ. правовая система.

5 Агальцов В. П., Титов В. М. Информатика для экономистов : учебник. – Москва : ИД «ФОРУМ» : ИНФРА-М, 2006. – 448 с.

6 Безмалый В. Мошенничество в Интернете // Компьютер пресс. – 2008. – №10. – С. 52-60.

7 Гафнер В. В. Информационная безопасность : учеб. пособие. – Ростов-на-Дону : Феникс, 2010. – 324 с.

8 Грибунин В. Г., Оков И. Н., Туринцев И. В. Цифровая стеганография. – Москва : Солон-Пресс, 2002. – 272 с.

9 Завгородний В. И. Комплексная защита информации в компьютерных системах : учеб. пособие. – Москва : Логос, 2001. – 264 с.

10 Компьютерные системы и сети : учеб. пособие / под ред. В. П. Косарева, Л. В. Еремина. – Москва : Финансы и статистика, 2001. – 464 с.

11 Ульянов В. Утечки конфиденциальной информации : профиль угроз URL : <http://compress.ru/articl.aspx?id=20513>.

12 Шалыгин В. Ф. Защита компьютерной информации. Эффективные методы и средства. – Москва : LRV Пресс, 2008. – 544 с.

7 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

7.1 ОСНОВЫ АЛГОРИТМИЗАЦИИ

7.1.1 Алгоритмы

Понятие алгоритма используется давно. Сам термин «алгоритм» произошел при переводе на европейские языки имени арабского математика IX века Аль-Хорезми, которым были описаны правила выполнения основных арифметических действий в десятичной системе счисления.

В зависимости от характера занятий в своей повседневной жизни люди встречаются с различными практическими задачами: приготовление кулинарного блюда по рецепту, проезд в общественном транспорте, поиск слова в словаре, переход улицы и т.д. При решении любой подобной задачи человек обращается к продуманным заранее со всеми возможными вариантами предписаниям о том, какие действия и в какой последовательности должны быть выполнены. Именно с целью успешного решения какого-либо определенного класса задач люди вырабатывают системы таких предписаний для использования разными людьми.

Алгоритм – это понятное и точное предписание исполнителю выполнить конечную последовательность действий, приводящих от исходных данных к искомому результату.

Задача составления алгоритма не имеет смысла, если не известны или не учитываются возможности его исполнителя, ведь выполнимость алгоритма зависит от того, какие действия может совершить исполнитель. Каждый исполнитель обладает определенным набором, системой команд исполнителя.

Но исполнителем алгоритмов может быть не только человек, но и автомат. Алгоритмом описывается работа любого механического устройства. В ряду всевозможных автоматов компьютер является лишь частным примером исполнителя, чье поведение реализуется на основе алгоритма. Более того, создание компьютеров оказало воздействие на развитие теории алгоритмов – одной из областей математики.

Алгоритм позволяет формализовать выполнение информационного процесса. Исполнитель может выполнять алгоритм формально, не вникая в содержание поставленной задачи, а только строго выполняя последовательность действий, предусмотренную алгоритмом. Исполнитель не обязан:

- понимать цель и методы достижения цели;
- пропускать или менять порядок действий;
- искать какую-то замену при невыполнимости этих действий.

Схема формальной работы исполнителя представлена на рисунке 7.1.

От разработчиков алгоритмов требуется знание и соблюдение правил составления алгоритмов. Эти правила заключаются в том, что алгоритм, предназначенный для исполнения автоматом, должен обладать следующими свойствами.



Рисунок 7.1 – Формальная работа исполнителя алгоритмов

Дискретность. Процесс решения задачи должен быть разбит на последовательность отдельных шагов. Таким образом формируется упорядоченная совокупность отделенных друг от друга команд (предписаний). Образованная структура алгоритма оказывается прерывной (дискретной): только выполнив одну команду, исполнитель сможет приступить к выполнению следующей.

Понятность. Алгоритм должен быть понятен исполнителю, и исполнитель должен быть в состоянии выполнить его команды. Следовательно, алгоритм нужно разрабатывать с ориентацией на определенного исполнителя, т.е. в алгоритм можно включать команды только из системы команд данного исполнителя.

Детерминированность. Будучи понятным, алгоритм не должен содержать команды, смысл которых может восприниматься неоднозначно. Например, команда «добавить по вкусу соли и перца» приведет к разным вариантам одного и того же блюда, так как «на вкус и цвет товарища нет». Кроме того, недопустимы ситуации, когда после выполнения очередной команды исполнителю не ясно, какую команду выполнять на следующем шаге. Нарушение составителем алгоритма этих требований приводит к тому, что одна и та же команда после выполнения разными исполнителями дает неодинаковый результат.

Результативность. Смысл этого свойства алгоритма состоит в том, что при точном исполнении всех команд алгоритма процесс решения задачи должен прекратиться за конечное число шагов и при этом должен быть получен определенный постановкой задачи ответ.

Массовость. Данное свойство означает, что алгоритм должен обеспечивать решение всего класса задач данного типа, т.е. может быть использован для любого набора исходных данных, допустимых данной задачей.

Однозначность. Данное свойство означает, что если применять один и тот же алгоритм к одному и тому же набору исходных данных, то каждый раз должен получаться один и тот же результат.

7.1.2 Формы представления и типы алгоритмов

Линейный тип алгоритма. Алгоритм, в котором команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий, является алгоритмом линейного типа. Таким будет, например, алгоритм вычислений по самым простейшим, безальтернативным формулам, не имеющим ограничений на значения входящих в них переменных.

Разветвляющийся тип алгоритма. В том случае, когда условие задачи

предусматривает в ходе её решения возможность выбора в зависимости от выполнения некоторых условий, алгоритм решения оказывается разветвляющимся. При организации ветвления необходимо помнить об одном важном свойстве этих алгоритмов: их исполнение всегда происходит по одному из возможных путей, который определяется конкретными текущими условиями, причем в каждом случае от начала алгоритма до его конца. Это свойство присуще всякому правильно составленному алгоритму и является признаком правильной организации ветвлений.

Циклический тип алгоритма. Алгоритм, составленный с использованием многократных повторений одних и тех же действий (циклов), называется циклическим. На практике чаще всего встречаются алгоритмы смешанного типа, у которых можно выделить блоки, имеющие структуру линейного, разветвляющегося или циклического типа.

Для записи алгоритма применяют разные формы – словесную, графическую, табличную, псевдокоды, программную.

Со словесной формой сталкивался каждый, кто хоть раз прибегал к помощи каких-либо инструкций по сборке какого-либо агрегата, по приготовлению какого-либо блюда. Алгоритм в этом случае записывается с помощью естественного языка.

Рассмотрим разные формы записи алгоритма вычисления площади круга по заданному радиусу.

Словесная форма

- 1 Прочитать значение R.
- 2 Умножить значение R на 3,14.
- 3 Умножить результат второго действия на значение R.
- 4 Записать полученный в предыдущей команде результат как значение S.

Табличная форма

R, см	$3,14 \times R$, см	$3,14 \times R \times R$, см ²
1	3,14	3,14
2	6,28	12,56

Табличная форма представления алгоритмов применяется только для линейных вычислительных алгоритмов.



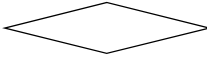


Псевдокоды – это полужформализованные описания алгоритмов на условном алгоритмическом языке, включающие как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.)

При использовании программной формы записи алгоритм записывается на языке программирования.

Графическая форма представления основана на замене типичных алгоритмических команд определенными алгоритмическими фигурами. Такое

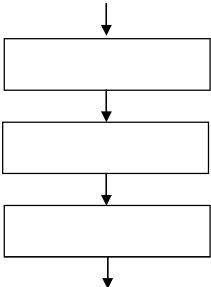
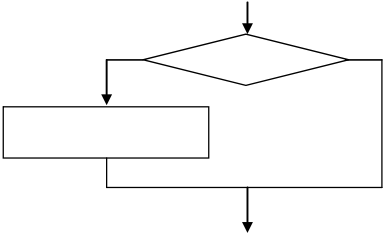
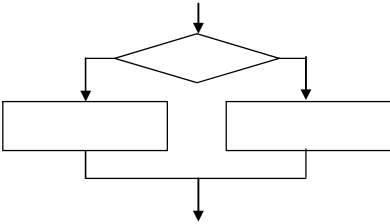
графическое представление называется схемой алгоритма или блок-схемой. В блок-схеме каждому типу действий соответствует геометрическая фигура, представленная в виде блочного символа. Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий. Для начертания этих схем используется набор символов, определяемых ГОСТ 19.701–90 (ИСО 5807–85) «Единая система программной документации». В таблице 7.1 приведены наиболее часто употребляемые символы.

Таблица 7.1 – Некоторые элементы записи блок-схем

Название элемента	Обозначение	Пояснение
Терминатор (пуск, останов)		Элемент отображает выход во внешнюю среду и вход из внешней среды (наиболее частое применение - начало и конец программы).
Процесс		Элемент отображает функцию обработки данных любого типа.
Решение		Отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа.
Данные (ввод-вывод)		Символ отображает данные. Носитель данных не определен.
Предопределенный процесс		Символ отображает выполнение процесса, состоящего из одной или нескольких операций или шагов программы, который определен в другом месте программы (в подпрограмме, модуле).

Алгоритм любой степени сложности можно построить с помощью блоков основного базового набора. Каждая из этих структур имеет только один вход и только один выход, что позволяет соединять между собой в процессе разработки алгоритма любое количество элементов базовых структур в любой последовательности. Базовые структуры представлены в таблице 7.2.

Таблица 7.2 – Базовые структуры

Псевдокод (Школьный алгоритмический язык)	Язык блок-схем
Следование	
действие 1 действие 2 ... действие n	
Псевдокод (Школьный алгоритмический язык)	Язык блок-схем
Ветвление в краткой форме	
если условие то действие все	
Ветвление в полной форме	
если условие то действие 1 иначе действие 2 все	
Цикл пока Предписывает выполнять тело цикла до тех пор, пока выполняется условие, записанное после слова пока	

<p>нц пока условие тело цикла(последовательность действий) кц</p>	
---	--

При записи алгоритма в словесной форме, в виде блок-схем или на псевдокоде допускаются определенные вольности при изображении команд. Вместе с тем такая запись точна настолько, что позволяет человеку понять суть дела и исполнить алгоритм.

Но помимо человека, исполнителем алгоритма выступают различные технические устройства и, в первую очередь, компьютер. Поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на понятном ему языке. Язык для записи алгоритмов называют языком программирования, а запись алгоритма на этом языке – программой для компьютера.

7.2 ИСТОРИЧЕСКИЙ АСПЕКТ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Рассматривая историю становления технологий разработки программ, выделяя основные этапы формирования программирования как науки, постараемся определить основные тенденции развития современных технологий программирования [1].

Фундамент к началу развития программирования был положен с появлением первых вычислительных машин. В тот период еще не сложилось какой-то технологии, разработка программ была искусством. Структура первых программ была достаточно простой (рисунок 7.2): описание операторов на машинном языке и обрабатываемые ими данные.

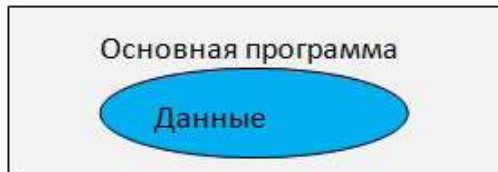


Рисунок 7.2 – Структура первых программ

С момента появления ассемблерных языков программы стали более понятными, т.к. стали использоваться символические имена данных и мнемоники кодов операций.

Увеличению сложности и упрощению написания программ способствовало появление языков высокого уровня (FORTRAN, ALGOL). В то же время становится возможным оперирование подпрограммами, записанными в отдельных файлах. Это способствовало созданию библиотек подпрограмм, которые по мере надобности вызывались из разрабатываемой программы.

Усложняется и структура программы, кроме основной программы и данных появляется набор подпрограмм, выполняющих обработку всех данных или их части (рисунок 7.3).



Рисунок 7.3 – Архитектура программы с глобальной областью данных

Архитектура программы с глобальной областью данных была несовершенной, т. к. с увеличением числа подпрограмм была высока вероятность искажения части глобальных данных какой-либо подпрограммой. Решение этой проблемы состояло в организации локальных данных для каждой подпрограммы (рисунок 7.4).

С начала 60-х годов назрел кризис в использовании подхода «снизу-вверх» при разработке программного обеспечения. Он был связан с попытками создания более сложных программ из набора более простых, что приводило к увеличению временных ресурсов на разработку. Это послужило толчком к развитию нового подхода к созданию сложных программных продуктов с целью снижения вероятности ошибок проектирования, который назван «структурным».

Второй этап становления программирования как науки (60-70-е годы XX в.) представляет собой совокупность разнообразных технологических приемов, направленных на выполнение всех этапов разработки программного продукта. Комплекс приемов основан на декомпозиции сложных систем с целью их реализации в виде совокупности отдельных подпрограмм (процедурная декомпозиция). Одновременно вводились ограничения на конструкции алгоритмов, ре-

комендовались формальные модели их описания, а также специальный метод проектирования алгоритмов – метод пошаговой детализации. Данный подход требовал представления одной задачи в виде иерархии подзадач более простой структуры.

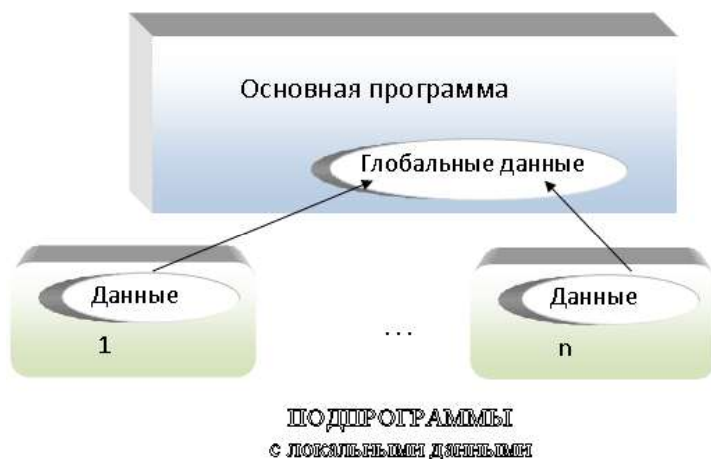


Рисунок 7.4 – Архитектура программы, использующей подпрограммы с локальными данными

Принцип структурного программирования лег в основу процедурных языков программирования (ALGOL-68, Pascal, C), которые поддерживали вложение подпрограмм, включали основные «структурные» операторы передачи управления, локализацию и ограничение области «видимости» данных.

Усложнение разрабатываемых программных продуктов и, как следствие, увеличение их размеров, потребовали структурирования данных. В результате в процедурных языках появляется возможность определения пользовательских типов данных. Стремление программистов к уменьшению количества ошибок, возникающих при работе с глобальными данными, дает начало развитию технологии модульного программирования.

В основе модульного программирования заключено выделение групп подпрограмм, использующих одни и те же глобальные данные в отдельно компилируемые модули (библиотеки подпрограмм), например, модуль графических ресурсов, модуль подпрограмм вывода на принтер (рисунок 7.5).

Между модулями существуют связи, поддерживаемые специальными интерфейсами, выполняется принцип инкапсуляции (доступ к «внутренним» переменным запрещен). Подобную технологию поддерживают современные версии языков Pascal и C (C++), языки Ада и Modula.

Модульное программирование позволяет вести разработку одного программного продукта несколькими программистами. Каждый разработчик реализует свой модуль независимо от остальных. Взаимодействие различных

модулей одной программы выполняется через специальные межмодульные интерфейсы (выбраны по согласованию с группой разработчиков).

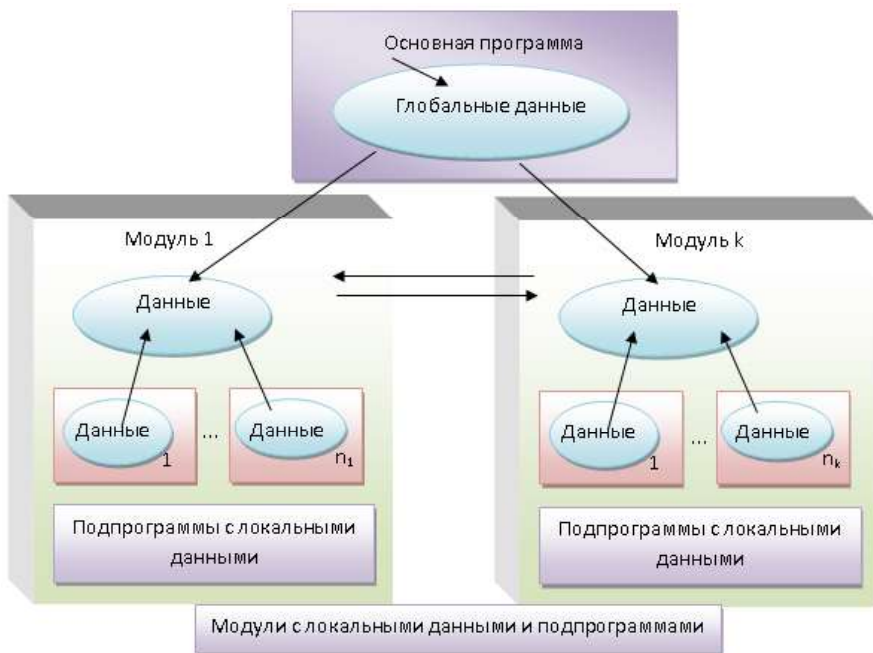


Рисунок 7.5 – Архитектура программы, состоящей из модулей

Разработанные модули в дальнейшем без изменений можно было использовать в других проектах, что повышало производительность труда программистов.

В дальнейшем модульные технологии стали прародителями объектного подхода к программированию (с середины 80-х до конца 90-х годов XX в.). Объектно-ориентированное программирование – это технология, в основе которой лежит представление программы как совокупности объектов, являющихся экземплярами класса. Взаимодействие программных объектов в такой системе осуществляется путем передачи сообщений (рисунок 7.6).

К достоинствам объектно-ориентированного программирования, в отличие от модульного программирования, можно отнести более естественную декомпозицию программного обеспечения, которая существенно облегчает его разработку. Программы, разрабатываемые средствами объектно-ориентированного подхода, основаны на механизмах наследования, полиморфизма, композиции, наполнения. Эти механизмы позволяют конструировать сложные объекты из сравнительно простых. В результате существенно увеличивается показатель повторного использования кодов, и появляется возможность создания библиотек классов для различных применений.

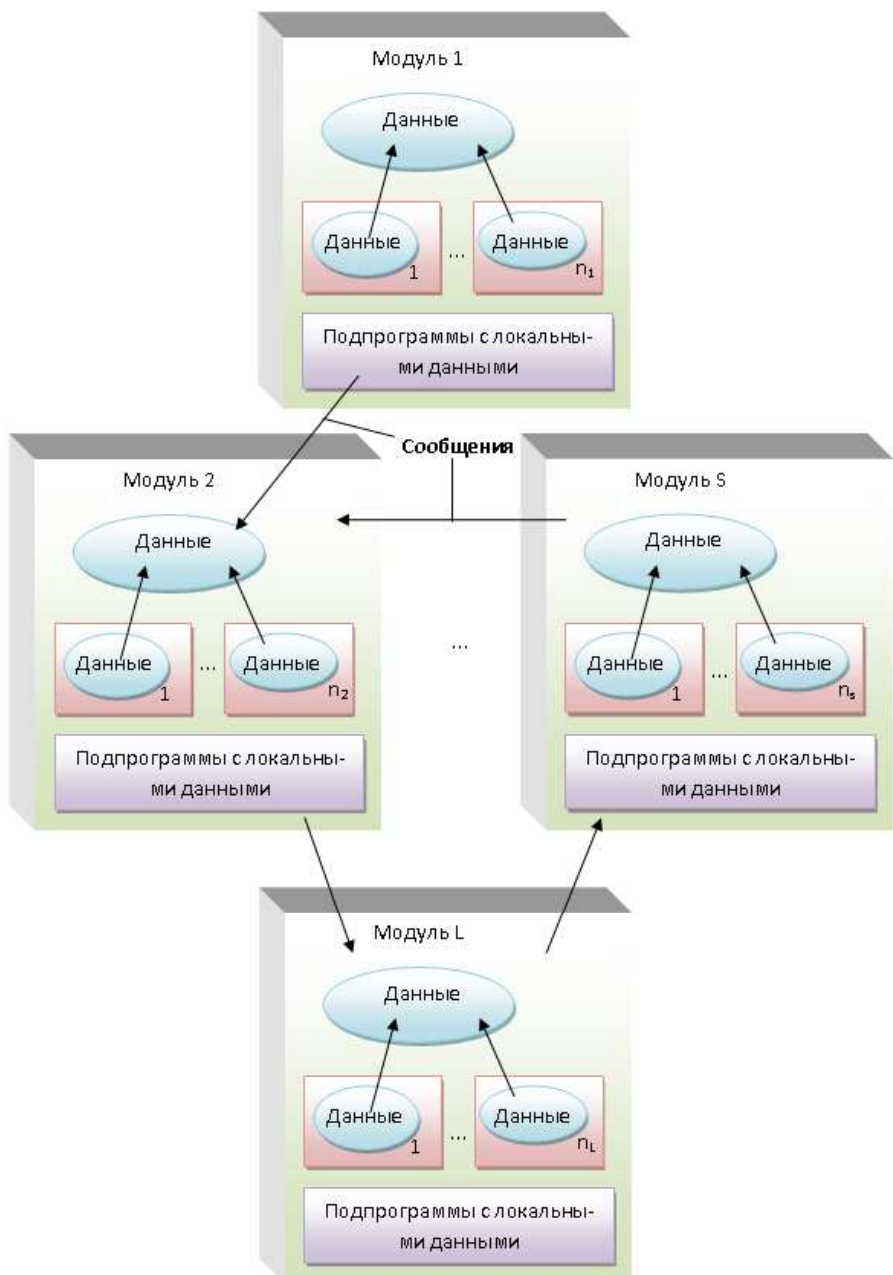


Рисунок 7.6 – Архитектура программы при объектно-ориентированном подходе

Следующим этапом развития технологий программирования было создание визуального программирования, например C++ Builder, Delphi, Visual C++ и др., что позволило разработчикам проектировать интерфейсы программного продукта с применением визуальных средств добавления и настройки специальных библиотечных компонентов. В результате визуального проектирования будет создана заготовка будущей программы, в которую уже внесены соответствующие коды.

С середины 90-х годов XX в. начинает развиваться совершенно новый компонентный подход к разработке программного обеспечения, предполагающий построение программного продукта из физически отдельно существующих частей (компонентов), взаимодействующих между собой через стандартизованные двоичные интерфейсы. Объекты-компоненты можно объединить в динамически вызываемые библиотеки или исполняемые файлы, распространять в двоичном виде (без исходных текстов) и использовать в любом языке программирования, поддерживающем соответствующую технологию.

Компонентный подход лежит в основе:

- технологии COM (Component Object Model – компонентная модель объектов):

√ определяет общую парадигму взаимодействия программ любых типов: библиотек, приложений, операционной системы;

√ позволяет одной части программного обеспечения использовать функции (службы), предоставляемые другой, независимо от того, функционируют ли эти части в пределах одного процесса, в разных процессах на одном компьютере или на разных компьютерах (рисунок 7.7);

- технология DCOM (Distributed COM – распределенная COM), созданная на базе технологии COM, решающая различные задачи разработки программного обеспечения;

- OLE-automation – технология создания программируемых приложений:

√ обеспечивает программируемый доступ к внутренним службам приложений;

√ вводит понятие диспинтерфейса – специального интерфейса, облегчающего вызов функций объекта. Например, Microsoft Excel, предоставляя другим приложениям свои службы, поддерживает эту технологию;

- ActiveX – технология, построенная на базе OLE-automation, предназначена для создания программного обеспечения как сосредоточенного на одном компьютере, так и распределенного в сети. Использует визуальное программирование для создания элементов управления ActiveX;

- MTS (сервер управления транзакциями) – технология, обеспечивающая безопасность и стабильную работу распределенных приложений при больших объемах передаваемых данных.

- MIDAS (сервер многозвенных распределенных приложений) – технология, организующая доступ к данным разных компьютеров с учетом балансировки нагрузки сети;

- технологии создания распределенных приложений CORBA (Common

Object Request Broker Architecture – общая архитектура с посредником обработки запросов объектов).

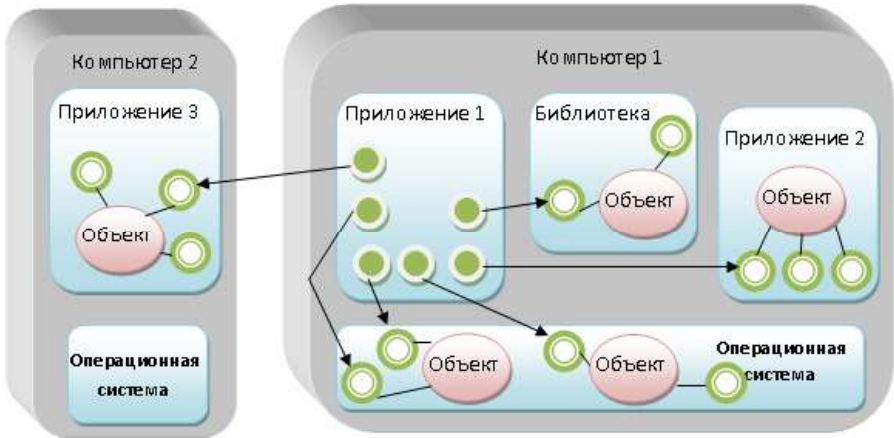


Рисунок 7.7 – Взаимодействие программных компонентов различных типов

Эти технологии используют сходные принципы и различаются лишь особенностями их реализации.

- технология CORBA, разработанная группой компаний OMG (группа внедрения объектной технологии программирования):

√ программное ядро реализовано для всех основных аппаратных и программных платформ;

√ эту технологию можно использовать для создания распределенного программного обеспечения в разнородной вычислительной среде;

√ организация взаимодействия между объектами клиента и сервера в CORBA осуществляется с помощью специального посредника, названного VisiBroker, и другого специализированного программного обеспечения.

Отличительной особенностью современного этапа развития технологии программирования, кроме изменения подхода, является создание и внедрение автоматизированных технологий разработки и сопровождения программного обеспечения, которые были названы CASE-технологиями (разработка программного обеспечения программных систем с использованием компьютерной поддержки).

7.3 Контрольные вопросы

1 Приведите примеры инструкций, не являющихся алгоритмом. Поясните – почему данные инструкции не являются алгоритмом.

2 Перечислите известные вам способы записи алгоритмов. Приведите примеры.

3 Что понимается под свойствами массовость, дискретность, понят-

ность. Приведите примеры.

4 Решение каких задач требует составления алгоритмов с ветвлением? Приведите примеры.

5 Решение каких задач требует составления алгоритмов с циклом? Приведите примеры.

6 Охарактеризуйте первый этап развития технологий программирования.

7 Охарактеризуйте второй этап развития технологий программирования.

8 Охарактеризуйте третий этап развития технологий программирования.

9 Охарактеризуйте четвертый этап развития технологий программирования.

Список источников

1 Гагарина Л. Г., Кокорева Е. В., Виснадул Б. Д. Технология разработки программного обеспечения : учеб. пособие / под ред. Л. Г. Гагариной. – Москва : ИД «ФОРУМ»: ИНФРА-М, 2008. – 400 с.

2 Информатика для юристов и экономистов : учебник для вузов / Симонич С. В. [и др.]. – 2-е изд. – СанктПетербург : Питер, 2013. – 544 с.

3 Лапчик М. П., Семакин И. Г., Хеннер Е. К. Методика преподавания информатики : учеб. пособие для студ. пед. вузов / под общ. ред. М. П. Лапчика. – 4-е изд., стер. – Москва : Изд. центр «Академия», 2007. – 624 с.

4 Шауцукова Л. З. Информатика : учеб. пособие для 10-11 кл. общеобразоват. учреждений. – Москва : Просвещение, 2004. – 416 с.

5 ГОСТ 19.701-90 (ИСО 5807- 85). ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. URL : http://snipov.net/c_4712_snip_100096.html .

8 МОДЕЛИРОВАНИЕ

8.1 ПОНЯТИЕ МОДЕЛИ

Слово *модель* от латинского слова *modulus* означает *образец, мера*. Первоначально значение слова *модель* было связано со строительным искусством. Во многих европейских языках моделью обозначался образ, прообраз или вещь, которая была похожей в каком-то отношении с другой вещью.

Понятие «модель» широко применяется в разных областях человеческой деятельности и может быть использовано в различных смысловых значениях. В моделировании рассматриваются такие модели, которые являются инструментами получения знаний.

Модель – это такой материальный или мысленно представляемый, т.е. информационный, объект, который в процессе исследования замещает объект-оригинал, обладая его существенными информационными свойствами (качественно-логическими и количественно-математическими), т.е. характером отношений между элементами изучаемого объекта и его отношений к другим объектам физической реальности, так, что изучение модели дает новые знания об объекте-оригинале. Более строго, по сути модель представляет собой вид информационной системы, копирующей целевые системы (информационные, энергетические, вещественные), и предназначенной для изучения свойств последних. По форме модель может быть воплощена на любом физическом носителе: вещественном изделии, компьютерной программе. [6]

Решаемая задача и средства ее решения определяют требования к модели. Существует ряд общих требований к моделям [3]:

- 1 адекватность – достаточно точное отображение свойств объекта;
- 2 полнота – предоставление получателю всей необходимой информации об объекте;
- 3 гибкость – возможность воспроизведения различных ситуаций во всем диапазоне изменения условий и параметров;
- 4 трудоемкость разработки должна быть приемлемой для имеющегося времени и программных средств.

В своей деятельности человек обычно создает и использует модели окружающего мира.

Они дают представление в наглядной форме объектов и процессов, недоступных для непосредственного восприятия, например:

- в физике – модели двигателей;
- в географии глобус – модель земли (реальный размер очень большой);
- в химии – модели кристаллических решеток, молекул (реальные размеры очень маленькие);
- в биологии – по муляжу человека изучаем внутреннее строение.

Один и тот же объект может иметь множество моделей, например, объект «ЧЕЛОВЕК». Его модели:

- в химии – БИОХИМИЧЕСКИЙ СОСТАВ;
- в анатомии – СКЕЛЕТ, СТРОЕНИЕ ВНУТРЕННИХ ОРГАНОВ;
- в физике – МАТЕРИАЛЬНАЯ ТОЧКА.

Разные объекты могут описываться одной моделью, например, модель «КАРТА». Её объекты:

- ПОЛЕЗНЫЕ ИСКОПАЕМЫЕ – на карте полезных ископаемых;
- КЛИМАТИЧЕСКИЕ ЗОНЫ – на карте климатических зон;
- ГОСУДАРСТВА, СТРАНЫ – на политической карте;
- ЗВЕЗДЫ – на звездной карте.

8.2 КЛАССИФИКАЦИЯ МОДЕЛЕЙ

Рассмотрим классификацию моделей по следующим признакам:

- 1 по области использования;
- 2 по фактору времени;
- 3 по отрасли знаний;
- 4 по форме представления.

По области использования модели можно разделить на:

1 Учебные модели – это модели, которые используются при обучении, например, тренажеры, обучающие программы;

2 Опытные модели – это модели, которые представляют собой увеличенные или уменьшенные копии объекта моделирования. Данные модели используются при исследовании и прогнозировании будущих характеристик объекта моделирования;

3 Научно-технические модели создаются и используются для исследования процессов и явлений, например, модель космического корабля;

4 Игровые модели – это модели, в которых проигрывается поведение объекта в различных условиях, например, экономические игры;

5 Имитационные модели отражают реальность с различной степенью точности, имитируя ее. Данные модели используют метод проб и ошибок. Примером имитационной модели может служить испытание побочных действий лекарственных препаратов.

По фактору времени модели делят на статические и динамические модели.

Статические модели – это такие модели, которые описывают состояние системы в определенный момент времени, т.е. единовременный срез информации по данному объекту. Например, строение молекул, список посаженных деревьев и т.д.

Динамические модели – это такие модели, которые описывают процессы изменения и развития системы, изменения объекта во времени. Примеры: описание движения тел, развития организмов, процесс химических реакций.

Модели по отрасли знаний классифицируют по отрасли деятельности человека. Модели бывают биологические, математические, социальные, химические, экономические, исторические и т.д.

По форме представления модели делятся на натурные и абстрактные.

Натурные или материальные модели – это предметные или физические модели. Они всегда реально реализуются. Материальные модели передают внешние свойства и внутреннее устройство исходных объектов, суть процессов и явлений объекта-оригинала. Данные модели строятся для экспериментального метода познания окружающей среды. Например, глобус, детские игрушки, скелет человека, макет солнечной системы, школьные пособия, физические и химические опыты.

Натурные модели можно разделить на:

- Физически-подобные модели. Это модели, которые сходны с оригиналом по физической природе и геометрической форме. Они отличаются от оригинала только числовыми значениями параметров, например, действующая модель электродвигателя;

- Пространственно-подобные модели. Это модели, которые сходны с оригиналом на основе физического подобия, такие, как макет самолета, ракеты;

- Математически-подобные модели. Это модели, которые не имеют с оригиналом ни физического, ни геометрического сходства, но модель и объект описываются одинаковыми уравнениями, например, аналогия между механическими и электрическими колебаниями.

Абстрактные, нематериальные или идеальные модели – это модели, которые не имеют реального воплощения. Их основу составляет информация. Это теоретический метод познания окружающей среды.

Выделяются следующие виды абстрактных моделей [8]:

- Вербальные (текстовые) модели. Эти модели используют последовательности предложений на формализованных диалектах естественного языка для описания той или иной области действительности.

Это модели образуются в результате раздумий, умозаключений. Они могут остаться мысленными или быть выражены словесно. Например, идея, возникшая в голове изобретателя, рифма в голове поэта.

- Математические модели – очень широкий класс знаковых моделей (основанных на формальных языках над конечными алфавитами), широко используемых те или иные математические методы.

Например, математическая модель звезды, представленная сложной системой уравнений, описывающих физические процессы, происходящие в недрах звезды; математические соотношения, позволяющие рассчитать оптимальный (наилучший с экономической точки зрения) план работы какого-либо предприятия.

Информационные модели – класс знаковых моделей, описывающих информационные процессы (возникновение, передачу, преобразование и использование информации) в системах самой разнообразной природы.

Рассмотрим типы информационных моделей:

- 1 Табличные модели содержат объекты и их свойства, которые описаны в виде списка, а их значения размещаются в ячейках прямоугольной формы. Перечень однотипных объектов размещается в первом столбце (или строке), а

значения их свойств – в следующих столбцах (или строках).

2 Иерархические модели содержат объекты, которые расположены по уровням. Каждый элемент высокого уровня состоит из элементов нижнего уровня, а элемент нижнего уровня может входить в состав только одного элемента более высокого уровня. Иерархическую модель можно представить в следующем виде (рисунок 8.1).

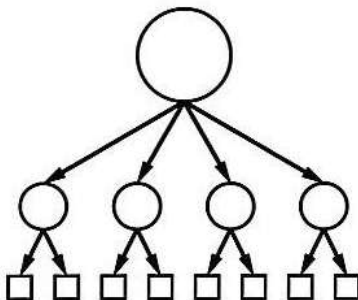


Рисунок 8.1 – Пример иерархической модели

3 Сетевые модели применяются для отражения систем, в которых связи между элементами имеют сложную структуру (рисунок 8.2).

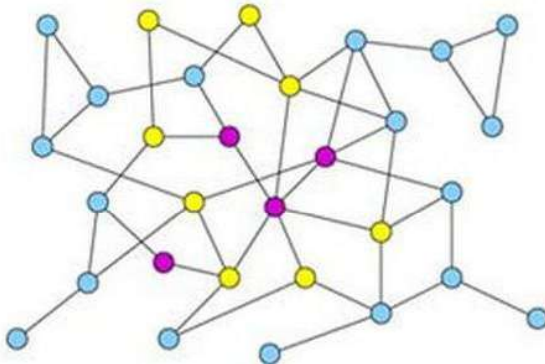


Рисунок 8.2 – Пример сетевой модели

По степени формализации информационные модели можно разделить на:

- 1 Образно-знаковые модели:
 - геометрические (рисунок, пиктограмма, чертеж, карта, план, объемное изображение);
 - структурные (таблица, граф, схема, диаграмма);
 - словесные (описание естественными языками);
 - алгоритмические (нумерованный список, пошаговое перечисление, блок-схема);

2 Знаковые модели:

- специальные – представлены на специальных языках (ноты, химические формулы);
- алгоритмические – программы.

8.3 ПОНЯТИЕ МОДЕЛИРОВАНИЯ

Под моделированием рассматривается процесс построения, изучения и применения моделей. Моделирование близко связано с такими категориями, как абстракция, аналогия, гипотеза и др., так как процесс моделирования включает в себя построение абстракций, умозаключений по аналогии, и конструирование научных гипотез. Модель выступает как инструмент познания, который помогает исследовать и изучить объект.

При построении модели исследователь отбрасывает те характеристики, параметры объекта-оригинала, которые несущественны для изучения объекта. Выбор характеристик объекта-оригинала, которые при этом сохраняются и входят в модель, определяется целями моделирования.

Естественные языки служат для создания описательных информационных моделей. В истории науки известны многочисленные описательные информационные модели. Например, гелиоцентрическая модель мира, которую предложил Коперник, формулировалась следующим образом [1]:

Земля вращается вокруг своей оси и вокруг Солнца; орбиты всех планет проходят вокруг Солнца.

С помощью формальных языков строятся формальные информационные модели (математические, логические и др.). Процесс построения информационных моделей с помощью формальных языков называется формализацией.

Процесс моделирования есть процесс перехода из реальной области в виртуальную (модельную) посредством формализации, далее происходит изучение модели (собственно моделирование) и, наконец, интерпретация результатов как обратный переход из виртуальной области в реальную. Этот путь заменяет прямое исследование объекта в реальной области, то есть лобовое или интуитивное решение задачи. Итак, в самом простом случае технология моделирования подразумевает три этапа: формализация, собственно моделирование, интерпретация (рисунок 8.3) [7].

Компьютерное моделирование – это метод решения задачи на основе использования ее компьютерной модели.

Суть компьютерного моделирования состоит в получении количественных и качественных результатов по построенной модели. Качественные выводы, которые получены по результатам анализа, позволяют найти неизвестные ранее свойства сложной системы: ее структуру, динамику развития, устойчивость, целостность и др. Количественные выводы в основном характеризуют прогноз некоторых будущих или объяснения прошлых значений переменных данной системы.



Рисунок 8.3 – Процесс моделирования (базовый вариант)

Рассмотрим основные этапы компьютерного моделирования:

- 1 Постановка задачи.
- 2 Разработка модели.
- 3 Компьютерный эксперимент.
- 4 Анализ результатов моделирования.

Опишем каждый из этих этапов.

Этап 1. Постановка задачи.

Этап постановки задачи характеризуется тремя основными стадиями: описание задачи, определение целей моделирования и формализация задачи.

Задача формулируется на обычном языке. Главное это - определить объект моделирования и понять, что собой должен представлять результат.

При этом нужно ответить на следующие вопросы:

- 1 Что дано по условию задачи?
- 2 Что требуется найти?
- 3 Какие данные допустимы?
- 4 Какие результаты и в каком виде должны быть получены, а какие нет?

По характеру постановки задачи можно разделить на две основные группы:

- «Что будет, если?» (цель моделирования в этом случае - исследовать изменение характеристик объекта при воздействии на него).

- «Как сделать, чтобы?» (цель моделирования в этом случае - какое произвести воздействие, чтобы параметры объекта удовлетворяли заданному условию?).

Формализацию проводят в виде поиска ответов на вопросы, уточняющие общее описание задачи.

Этап 2. Разработка модели

При построении модели сначала определяется вид модели:

1 Математическая модель. При построении данной модели выполняется формальная (математическая) постановка задачи, т.е. представление ее в виде уравнений, соотношений, ограничений и т.п.

2 Информационная модель. При построении данной модели выясняются свойства, состояния, действия и другие характеристики элементарных объектов в любой форме: устно, в виде схем, таблиц. Формируется представление об элементарных объектах, составляющих исходный объект, т. е. информационная модель. Модели должны отражать наиболее существенные признаки, свойства, состояния и отношения объектов предметного мира. Именно они дают полную информацию об объекте.

3 Компьютерная модель. Это модель, реализованная средствами программной среды. От выбора программной среды зависит алгоритм построения компьютерной модели, а также форма его представления. В среде программирования – это программа, записанная на языке программирования. В прикладных средах – это последовательность технологических приемов, приводящая к решению задачи.

Помимо этих моделей используются и другие виды моделей.

Этап 3. Компьютерный эксперимент

На данном этапе происходит тестирование модели.

Тест – набор исходных данных, позволяющих определить правильность построения модели. При подборе тестов следует предусмотреть:

1 проверку основных частных случаев;

2 проверку основных типов недопустимых данных;

3 проверку пограничных условий, т.е. тех, которые лежат на границе допустимых и недопустимых значений.

Этап 4. Анализ результатов моделирования

Конечная цель моделирования – принятие решения, которое должно быть выработано на основе всестороннего анализа результатов моделирования. Этот этап решающий – либо продолжается исследование, либо заканчивается. Возможно, известен ожидаемый результат, тогда необходимо сравнить полученный и ожидаемый результаты. В случае совпадения можно принять решение.

Основой для выработки решения служат результаты тестирования и экспериментов. Если результаты не соответствуют целям поставленной задачи, значит, были допущены ошибки на предыдущих этапах. Это может быть слишком упрощенное построение информационной модели, либо неудачный выбор метода или среды моделирования, либо нарушение технологических приемов при построении модели. Если такие ошибки выявлены, то требуется корректировка модели, т. е. возврат к одному из предыдущих этапов. Процесс повторяется до тех пор, пока результаты эксперимента не будут отвечать целям моделирования.

Рассмотрим этапы построения компьютерной модели на примере следующей задачи. Расстояние по реке равно S км. Лодка проходит этот путь по течению за t_1 ч, а против течения за t_2 ч. Найдите собственную скорость лодки

и скорость течения реки.

Этап 1. Постановка задачи и ее уточнение.

Описание задачи

1 Что дано по условия задачи?

Дано: S – расстояние по реке

t_1 – время, за которое лодка проходит по течению реки

t_2 - время, за которое лодка проходит против течения реки

2 Что требуется найти?

Найти: v_n – скорость лодки по течению реки

v_p – скорость реки

3 Какие данные допустимы?

$S > 0, t_1 > 0, t_2 > 0, t_2 > t_1$

4 Какие результаты и в каком виде должны быть получены, а какие нет?

$v_n \geq 0, v_p > 0$

Цель моделирования

Определить движение лодки по реке

Этап 2. Разработка модели.

Математическая модель.

Данная задача сводится к решению следующей системы уравнений:

$$\begin{cases} S = t_1(v_n + v_p) \\ S = t_2(v_n - v_p) \end{cases}$$

Выразим из первого уравнения v_n .

$$v_n = \frac{S}{t_1} - v_p.$$

Подставим во второе уравнение, получим:

$$S = t_2 \left(\frac{S}{t_1} - v_p - v_p \right).$$

Выразим из уравнения v_p

$$\frac{S}{t_2} = \frac{S}{t_1} - 2v_p;$$

$$v_p = \frac{1}{2} \left(\frac{S}{t_1} - \frac{S}{t_2} \right).$$

Подставим полученную v_p в выраженное v_n . Получим

$$v_n = \frac{S}{t_1} - \frac{1}{2} \left(\frac{S}{t_1} - \frac{S}{t_2} \right).$$

Этап 3. Разработка алгоритма.

1 ввести S, t_1, t_2 ;

- 2 проверить правильно ли введены (на ограничения);
- 3 произвести расчет v_d, v_p ;
- 4 вывести v_d, v_p .

Этап 4. Компьютерная реализация.

Реализовать можно при помощи любого языка программирования либо любого программного комплекса, который позволяет проводить исследование информационных моделей.

Этап 5. Тестирование (таблица 8.1):

Таблица 8.1– Тестирование

S	t_1	t_2	v_d	v_p
0	0	0	Недопустимые данные	
-4	5	4	Недопустимые данные	
6	-5	6	Недопустимые данные	
5	7	-7	Недопустимые данные	
10	5	9	Недопустимые данные	
20	2	5	7	3
20	4	2	Недопустимые данные	

8.4 ГРАФИЧЕСКИЕ ИНФОРМАЦИОННЫЕ МОДЕЛИ

Графические информационные модели представляют некоторый объект в общих, главных чертах с помощью условных обозначений. С их помощью может быть представлен и внешний вид объекта, и его структура. При этом модель не претендует на полноту предоставления информации об объекте. Примерами графических моделей могут служить различные схемы, карты, чертежи, графики и диаграммы (рисунок 8.4).

8.5 ГРАФЫ

На рисунке 8.5 изображена схема дорог между пятью населёнными пунктами А, В, С, D, Е. На данной схеме населённые пункты обозначены буквами, а дороги – линиями. Это не карта местности. На этой схеме отражен лишь факт существования пяти поселков и дорожной связи между ними. Такая схема, отображающая элементный состав системы и структуру связей, называется *графом*.

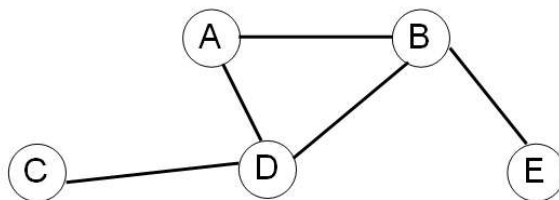


Рисунок 8.5 – Схема дорог

Составными частями графа являются вершины и ребра. Вершины графа могут изображаться кругами, овалами, точками, прямоугольниками и т. д. Вершины графа обозначают элементы системы, а ребра показывают связи (отношения) между элементами.

Граф называется *взвешенным*, если его вершины или ребра имеют некоторые количественные характеристики – веса вершин или ребер.

На рисунке 8.6 с помощью взвешенного графа изображены дороги между населёнными пунктами; веса ребер – это протяжённость дорог в километрах. В качестве веса вершин можно указать, например, численность население каждого поселка.

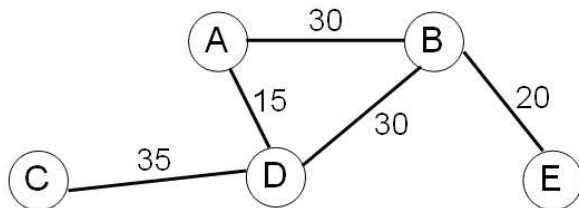


Рисунок 8.6 – Взвешенный граф

Путь по вершинам и ребрам графа, в который любое ребро графа входит не более одного раза, называется *цепью*. Цепь, начальная и конечная вершины которой совпадают, называется *циклом*. На рисунке 8.6 циклом соединяются населенные пункты А, В и D.

Граф с циклом называется *сетью*. Для сети характерна возможность множества различных путей перемещения по ребрам между некоторыми парами вершин.

На рисунках 8.5 и 8.6 каждое ребро обозначает наличие дорожной связи между двумя пунктами. При этом если дорожная связь действует одинаково в обе стороны, то по дороге можно проехать как от А к В, так и от В к А. Такие графы являются *неориентированными*, а их связи называют *симметричными*.

Если допустить, что между населенными пунктами А, В и D разрешено только одностороннее движение, то получим другой пример графа (рисунок 8.7).

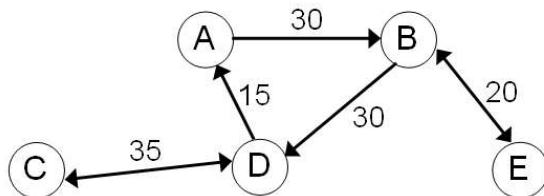


Рисунок 8.7 – Взвешенный граф

Связи между вершинами данного графа *несимметричны* и поэтому изображаются направленными линиями со стрелками. Такие линии принято называть *дугами* (в отличие от ребер неориентированных графов). Граф с такими

свойствами называется *ориентированным*. Линия, выходящая и входящая в одну и ту же вершину, называется *петлей*. В данном примере присутствуют две петли: С – D и В – E.

8.6 ДЕРЕВЬЯ

При построении информационных моделей многих систем часто приходится иметь дело с иерархической структурой. Иерархическая структура естественным образом возникает, когда объекты или некоторые их свойства находятся в отношении соподчинения (вложения, наследования).

Как правило, иерархическую структуру имеют системы административного управления, между элементами которых установлены отношения подчиненности. Иерархическую структуру имеют также системы, между элементами которых существуют отношения вхождения одних в другие.

На рисунке 8.8 изображен граф, отражающий иерархическую административную структуру университета: университет делится на факультеты; факультеты делятся на учебные группы, в состав которых входят студенты.

Граф иерархической структуры называется *деревом*. Основным свойством дерева является то, что между любыми двумя его вершинами существует единственный путь. Деревья не содержат циклов и петель.

Всякая иерархическая система может быть представлена с помощью дерева. У дерева выделяется одна главная вершина, называемая его *корнем*. Эта вершина изображается вверху; от нее идут ветви дерева. От корня начинается отсчет уровней дерева. Вершины, непосредственно связанные с корнем, образуют первый уровень. От них идут связи к вершинам второго уровня и т. д. Каждая вершина дерева (кроме корня) имеет одну исходную вершину на предыдущем уровне и может иметь множество порожденных вершин на следующем уровне. Такой принцип связи называется «*один ко многим*». Вершины, не имеющие порожденных вершин, называются *листьями*.



Рисунок 8.8 – Иерархическая структура университета

Иерархическими являются различные системы классификации в науке и технике. Например, система хранения файлов на магнитных дисках организована по иерархическому принципу. Операционная система позволяет получить на экране компьютера изображение файловой структуры в виде дерева (рису-

нок 8.9). Корнем этого дерева является корневой каталог диска, вершинами – подкаталоги разных уровней. Путь к файлу – это путь от корневого каталога до каталога, непосредственно содержащего данный файл. Для каждого файла такой путь единственный. Например, путь к файлам, содержащимся в каталоге DOC2 (рисунок 8.9), описывается так: \WORK\DOCUMENT\DOC2\.

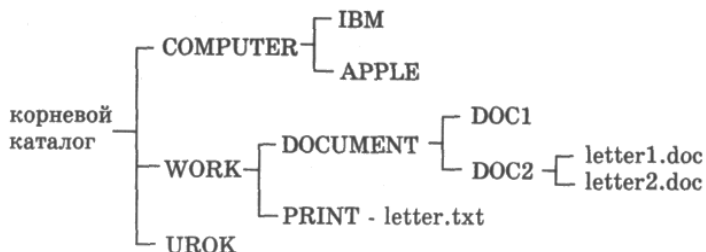


Рисунок 8.9 – Иерархическая структура хранения файлов на диске

При поиске файлов перемещение по дереву может происходить только вверх или вниз, при этом нельзя выполнить прямой переход между вершинами одного уровня.

Графы как информационные модели широко применяются во многих сферах человеческой деятельности. Например, можно дома и сооружения в населенном пункте изобразить вершинами, а соединяющие их дороги - рёбрами графа. Такой граф позволит спланировать оптимальную схему движения транспортных маршрутов, кратчайшие объездные пути, расположение торговых точек и других объектов.

8.7 ИСПОЛЬЗОВАНИЕ ГРАФОВ ПРИ РЕШЕНИИ ЗАДАЧ

Рассмотрим следующую игру: сначала в кучке лежат 5 камней; два игрока убирают камни по очереди, причём за один ход можно убрать 1 или 2 камня; выигрывает тот, кто оставит в кучке 1 камень. Требуется определить победителя в игре.

Первый игрок может убрать 1 камень или 2 камня. В первом случае в куче останется 4 камня, а во втором – 3 камня.

Если после хода первого игрока в куче осталось 4 камня, то второй игрок может одним своим ходом оставить 2 или 3 камня. Если же после хода первого игрока осталось 3 камня, то второй игрок может выиграть, взяв два камня и оставив один.

Если после второго игрока осталось 3 или 2 камня, то первый игрок в любом случае имеет шанс на выигрыш.

Таким образом, при правильной стратегии игры всегда выигрывает первый игрок, если своим первым ходом он возьмет один камень.

На рисунке 8.10 представлен граф, называемый деревом игры. На нём представлены все возможные варианты ходы игроков, в том числе пунктирными линиями обозначены ошибочные (проигрышные) ходы.

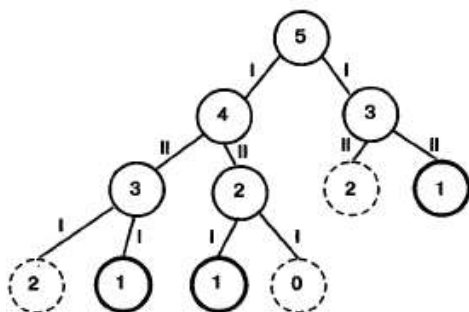


Рисунок 8.10 – Дерево игры

8.8 Контрольные вопросы

- 1 Что понимают под моделью?
- 2 Перечислите требования, предъявляемые к модели.
- 3 Приведите примеры моделей.
- 4 Перечислите признаки, по которым можно классифицировать модель.
- 5 Какие модели выделяют по области использования?
- 6 Какие модели выделяют по фактору времени?
- 7 Какие модели выделяют по форме представления?
- 8 Какие модели выделяют по степени формализации?
- 9 Что понимают под моделированием?
- 10 Перечислите основные этапы компьютерного моделирования.
- 11 Приведите примеры материальных моделей, не упомянутых в тексте раздела.
- 12 Что такое информационная модель? Приведите примеры информационных моделей, не упомянутых в тексте раздела.
- 13 Какие модели называют вербальными?
- 14 Какие информационные модели относят к графическим?
- 15 Что такое граф?
- 16 Каков элементный состав графа? Охарактеризуйте каждый тип элементов.
- 17 Приведите пример системы, модель которой можно представить в форме графа.
- 18 Что такое сеть?
- 19 Какой граф называют ориентированным?
- 20 Что представляет собой иерархическая структура? Где она применяется?

Список источников

1 Беляев М. А., Лысенко В. В., Малинина Л. А. Основы информатики : учебник для вузов. – Ростов-на-Дону : Феникс, 2006. URL: <http://wm-help.net/lib/b/book/120467185/>.

2 Босова Л. Л. Информатика : учебник для 9 кл. (в двух частях). – Москва : БИНОМ. Лаборатория знаний, 2012.

3 Васильев К. К., Служивый М. Н. Математическое моделирование систем связи : учеб. пособие. – Ульяновск : УлГТУ, 2008. URL: http://sernam.ru/book_mm.php.

4 Замятина О. М. Моделирование систем : учеб. пособие. – Томск : Изд-во ТПУ, 2009. URL : <http://window.edu.ru/catalog/pdf2txt/826/74826/54902>.

5 Клюев С. А. Компьютерное моделирование : учеб.-метод. пособие. – Волжский : Изд-во ВПИ, 2009. URL : http://window.edu.ru/catalog/pdf2txt/488/61488/31246?p_page=1.

6 Леонова Н. Л. Компьютерное моделирование : курс лекций. – Санкт-Петербург, 2015. URL : <http://simulation.su/uploads/files/default/2015-kurs-lection-leonova-1.pdf>.

7 Оболонин И. А. Основы компьютерного проектирования и моделирования РЭС : учеб. пособие. – Новосибирск, 2012. URL : <http://www.studfiles.ru/preview/2930789/>.

8 Шеннон Р. Имитационное моделирование систем – искусство и наука. – Москва : Мир, 1978. – 418 с.

СОДЕРЖАНИЕ

1 ИНФОРМАТИКА И ИНФОРМАЦИЯ.....	3
1.1. Информатика.....	3
1.1.1 Предмет и задачи информатики.....	3
1.1.2 Истоки и предпосылки информатики.....	4
1.2 Информация и информационные процессы.....	5
1.2.1 Различные подходы к понятию «информация».....	5
1.2.2 Свойства информации.....	7
1.2.3 Измерение информации.....	11
1.2.4 Единицы измерения информации.....	13
1.2.5 Информационные процессы.....	14
1.3 Кодирование информации.....	16
1.3.1 Кодирование числовой информации.....	16
1.3.2 Кодирование символьной информации.....	19
1.3.3 Кодирование графической информации.....	20
1.3.4 Кодирование звуковой информации.....	23
1.4 Контрольные вопросы.....	25
Список источников.....	25
2 ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА.....	27
2.1 История развития вычислительной техники.....	27
2.1.1 Нулевое поколение – механические компьютеры (1642-1945).....	27
2.1.2 Первое поколение – электронные лампы (1945-1955).....	29
2.1.3 Второе поколение – транзисторы (1955-1965).....	31
2.1.4 Третье поколение – интегральные схемы (1965-1980).....	33
2.1.5 Четвертое поколение – сверхбольшие интегральные схемы (1980).....	34
2.1.6 Пятое поколение – компьютеры небольшой мощности и невидимые компьютеры.....	35
2.2 Методы классификации компьютеров.....	36
2.2.1 Классификация по назначению.....	36
2.2.2 Классификация по уровню специализации.....	37
2.2.3 Классификация по типоразмерам.....	38
2.2.4 Классификация по совместимости.....	39
2.3 Аппаратная конфигурация персонального компьютера.....	39
2.3.1 Базовая конфигурация персонального компьютера.....	39
2.3.2. Внутренние устройства системного блока.....	44
2.3.3 Системы, расположенные на материнской плате.....	50
2.4 Контрольные вопросы.....	53
Список источников.....	54
3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ.....	55
3.1 Классификация ПО ЭВМ.....	55

3.2 Системное программное обеспечение.....	55
3.2.1 Операционная система: понятие, основные функции, структура	56
3.2.2 Архитектура ОС	57
3.2.3 Классификация ОС и краткий обзор	59
3.2.4 Файловая система: основные понятия, функции	60
3.2.5 Сервисные программы.....	63
3.3 Прикладное программное обеспечение.....	65
3.3.1 Прикладное ПО общего назначения	66
3.3.2 Проблемно-ориентированное прикладное ПО.....	67
3.3.3 Интегрированные системы	69
3.4 Инструментальное программное обеспечение	69
3.4.1 Трансляция программ: основные понятия, этапы трансляции	70
3.4.2 Структура системы программирования	71
3.5 Распространение программного обеспечения	72
3.6 Контрольные вопросы.....	72
Список источников	73
4 КОМПЬЮТЕРНЫЕ СЕТИ, ИНТЕРНЕТ И МУЛЬТИМЕДИА	
ТЕХНОЛОГИИ	74
4.1 Понятие компьютерной сети	74
4.2 Классификация вычислительных сетей	76
4.3 Логическая топология и основное коммуникационное оборудование	
вычислительных сетей	80
4.4 Появление и развитие сети Интернет	82
4.5 Понятие мультимедиа	84
4.6 Аудиоинформация	85
4.7 Видеоинформация	86
4.8 Контрольные вопросы.....	88
Список источников	88
5 ОСНОВЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ.....	90
5.1 Базы данных. Банки данных. СУБД.....	90
5.1.1 Понятие предметной области.....	90
5.1.2 Модели данных.....	90
5.1.3 Базы данных. Классификация баз данных	91
5.1.4 Банки данных. Базы знаний. Отличие баз данных от баз знаний	94
5.1.5 Классификация систем управления базами данных	95
5.1.6 Классификация информационных систем	97
5.2 Основные концепции реляционных баз данных	98
5.2.1 Реляционная модель данных.....	98
5.2.2 Поле и запись таблицы	98
5.2.3 Первичный и внешний ключи таблицы	99
5.2.4 Ссылочная целостность.....	102

5.3	Этапы проектирования баз данных.....	103
5.3.1	Основные этапы проектирования баз данных.....	103
5.3.2	Первый этап – анализ предметной области.....	104
5.3.3	Второй этап – этап логического проектирования.....	110
5.3.4	Третий этап – этап физического проектирования.....	113
5.4	Контрольные вопросы.....	114
	Список источников.....	115
6	КОНТРОЛЬ И ЗАЩИТА ИНФОРМАЦИИ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ.....	117
6.1	Основные понятия защиты информации.....	117
6.2	Угрозы безопасности информации в компьютерных системах.....	119
6.2.1	Виды случайных угроз.....	119
6.2.2	Виды преднамеренных угроз.....	121
6.3	Комплексные методы защиты информации.....	125
6.3.1	Правовые методы защиты.....	125
6.3.2	Организационные методы защиты.....	128
6.3.3	Криптографические методы защиты.....	129
6.3.4	Стеганографические методы защиты.....	132
6.4	Контрольные вопросы.....	133
	Список источников.....	133
7	ОСНОВЫ АЛГОРИТМИЗАЦИИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ.....	135
7.1	Основы алгоритмизации.....	135
7.1.1	Алгоритмы.....	135
7.1.2	Формы представления и типы алгоритмов.....	136
7.2	Исторический аспект технологии программирования.....	140
7.3	Контрольные вопросы.....	146
	Список источников.....	147
8	МОДЕЛИРОВАНИЕ.....	148
8.1	Понятие модели.....	148
8.2	Классификация моделей.....	149
8.3	Понятие моделирования.....	152
8.4	Графические информационные модели.....	157
8.5	Графы.....	157
8.6	Деревья.....	159
8.7	Использование графов при решении задач.....	160
8.8	Контрольные вопросы.....	161
	Список источников.....	161

Учебное издание

Информатика. Базовый курс

Учебное пособие

Редактор Е.В. Денисенко

Подписано в печать 11.12.17	Формат 60x84 1/16	Бумага 80 г/м ²
Печать цифровая	Усл. печ. л. 10.38	Уч. изд.л. 10.38
Заказ № 230	Тираж 100	

Библиотечно-издательский центр КГУ.
640020, г. Курган, ул.Советская, 63/4.
Курганский государственный университет.