

ПРЕДИСЛОВИЕ¹

Интернет-версия пособия "Информатика" состоит из двух разделов:

- Теория (с задачами и решениями);
- Практикум по алгоритмизации и программированию.

Теоретический раздел представляет собой попытку создания на доступном уровне цельной картины курса информатики в фундаментальном его аспекте. В нем рассматриваются такие содержательные линии курса информатики, как информация и информационные процессы, представление информации, компьютер, алгоритмы и исполнители, моделирование и формализация.

Объем бумажного пособия не позволил осветить лишь линию стремительно обновляющихся информационных технологий, по которой регулярно выходят хорошие пособия.

Материал раздела распределен по главам и теоретическим пунктам в форме вопросов и ответов, включает специально подобранные оригинальные примеры, задачи и упражнения, выполненные с применением и анализом различных методических и технологических приемов.

Каждая глава заканчивается большим количеством задач и упражнений для самостоятельного решения, для которых приведены ответы, указания и образцы выполнения.

Практикум по алгоритмизации и программированию предназначен для развития навыков алгоритмического мышления и обучения основам программирования. Он ориентирован на учащихся, имеющих начальное представление об орфографии языков Turbo Pascal.

Известно, что после ознакомления с основами какого-либо алгоритмического языка учащемуся необходимо выполнить большое количество развивающих упражнений, а затем разобрать и самостоятельно составить сотни разнообразных алгоритмов и реализующих их программ.

Практикум, содержащий множество подробно откомментированных характерных примеров и сотни задач, последовательно и целенаправленно вовлекает учащегося в процесс самостоятельного и осмысленного составления законченных программ. Вырабатывает необходимые составляющие алгоритмической и программистской грамотности:

- ясный и понятный стиль,
- надежность и эффективность решений,
- умение организовать переборы и ветвления и т.п.

Для облегчения усвоения курса и повышения эффективности обучения учебный материал

¹ Файлы созданы на основе интернет-версии издания: *Шауцукова Л.З. Информатика 10 - 11. — М.: Просвещение, 2000 г* (<http://www.tomsk.ru/Books/informatica/theory/index.html>)

практикума подан в нетрадиционной, ориентированной на общность алгоритмических конструкций компоновке.

Практикум универсален в том отношении, что позволяет выработать полноценные навыки алгоритмизации и программирования независимо от качества компьютерного оснащения учебного заведения или при полном отсутствии такового. Для этого каждый алгоритм дается в следующей последовательности:

- формулировка задачи;
- система тестовых данных и результатов;
- реализация алгоритма в четырех наиболее популярных в школьном образовании средах — на школьном алгоритмическом языке, на языке блок-схем и на языках программирования Turbo Pascal;
- таблицы исполнения алгоритма на каждом из тестов.

Для многих задач приводятся результаты работы программ, выведенные на экран.

Важное значение, придаваемое тестированию алгоритмов, объясняется тем, что на этом этапе

- детально изучается и уточняется условие задачи;
- происходит осмысление того, что является исходными данными и результатами;
- фиксируются все ситуации, которые могут возникнуть при решении задачи;
- уточняются типы данных;
- даются имена переменным;
- продумываются формы представления и выдачи исходных данных и результатов.

Приводимые способы и программы решения задач по возможности являются рациональными, но не претендуют на то, чтобы быть наилучшими. Так, в программах с целью экономии объема не предусмотрена защита от недопустимых данных, хотя это обязательный элемент любой программы. Читатель может сам восполнить эти недочеты, воспользовавшись рекомендациями восьмой главы первого раздела, и в ряде случаев предложить более совершенное решение задачи.

Глава 1. Введение в информатику

1.1. Что такое информатика?

Термин "**информатика**" (франц. *informatique*) происходит от французских слов *information* (информация) и *automatique* (автоматика) и дословно означает "**информационная автоматика**".

Широко распространён также англоязычный вариант этого термина — "**Computer science**", что означает буквально "**компьютерная наука**".

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 году 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 — комплексная научная дисциплина с широчайшим диапазоном применения. Её **приоритетные направления**:

- **разработка вычислительных систем и программного обеспечения;**
- **теория информации**, изучающая процессы, связанные с передачей, приёмом, преобразованием и хранением информации;
- **математическое моделирование, методы вычислительной и прикладной математики и их применение к фундаментальным и прикладным исследованиям в различных областях знаний;**
- **методы искусственного интеллекта**, моделирующие методы логического и аналитического мышления в интеллектуальной деятельности человека (логический вывод, обучение, понимание речи, визуальное восприятие, игры и др.);
- **системный анализ**, изучающий методологические средства, используемые для подготовки и обоснования решений по сложным проблемам различного характера;
- **биоинформатика**, изучающая информационные процессы в биологических системах;
- **социальная информатика**, изучающая процессы информатизации общества;
- **методы машинной графики, анимации, средства мультимедиа;**
- **телекоммуникационные системы и сети**, в том числе, **глобальные компьютерные сети**, объединяющие всё человечество в единое информационное сообщество;

- **разнообразные приложения**, охватывающие производство, науку, образование, медицину, торговлю, сельское хозяйство и все другие виды хозяйственной и общественной деятельности.

Российский академик А.А. Дородницын выделяет в информатике три неразрывно и существенно связанные части — **технические средства, программные и алгоритмические**.

Технические средства, или аппаратура компьютеров, в английском языке обозначаются словом **Hardware**, которое буквально переводится как "твердые изделия".

Для обозначения **программных средств**, под которыми понимается **совокупность всех программ, используемых компьютерами, и область деятельности по их созданию и применению**, используется слово **Software** (буквально — "мягкие изделия"), которое подчеркивает равнозначность самой машины и программного обеспечения, а также способность программного обеспечения модифицироваться, приспосабливаться и развиваться.

Программированию задачи всегда предшествует **разработка способа ее решения в виде последовательности действий, ведущих от исходных данных к искомому результату**, иными словами, **разработка алгоритма решения задачи**. Для обозначения части информатики, связанной с разработкой алгоритмов и изучением методов и приемов их построения, применяют термин **Brainware** (англ. brain — интеллект).

Роль информатики в развитии общества чрезвычайно велика. С ней связано начало революции в области накопления, передачи и обработки информации. Эта революция, следующая за революциями в овладении веществом и энергией, затрагивает и коренным образом преобразует не только сферу материального производства, но и интеллектуальную, духовную сферы жизни.

Прогрессивное увеличение возможностей компьютерной техники, развитие информационных сетей, создание новых информационных технологий приводят к значительным изменениям во всех сферах общества: в производстве, науке, образовании, медицине и т.д.

1.2. Что такое информация?

Термин "**информация**" происходит от латинского слова "**informatio**", что означает **сведения, разъяснения, изложение**. Несмотря на широкое распространение этого термина, понятие информации является одним из самых дискуссионных в науке. В настоящее время наука пытается найти общие свойства и закономерности, присущие многогранному понятию *информация*, но пока это понятие во многом остается интуитивным и получает различные смысловые наполнения в различных отраслях человеческой деятельности:

- **в обиходе** информацией называют любые данные или сведения, которые кого-либо интересуют. Например, сообщение о каких-либо событиях, о чьей-либо

деятельности и т.п. *"Информировать"* в этом смысле означает *"сообщить нечто, неизвестное раньше"*;

- **в технике** под информацией понимают сообщения, передаваемые в форме знаков или сигналов;
- **в кибернетике** под информацией понимает ту часть знаний, которая используется для ориентирования, активного действия, управления, т.е. в целях сохранения, совершенствования, развития системы (Н. Винер).

Клод Шеннон, американский учёный, заложивший основы теории информации — науки, изучающей процессы, связанные с передачей, приёмом, преобразованием и хранением информации, — **рассматривает информацию как снятую неопределенность наших знаний о чем-то.**

Приведем еще несколько определений:

- *Информация — это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний* (Н.В. Макарова);
- *Информация — это отрицание энтропии* (Леон Бриллюэн);
- *Информация — это мера сложности структур* (Моль);
- *Информация — это отраженное разнообразие* (Урсул);
- *Информация — это содержание процесса отражения* (Тузов);
- *Информация — это вероятность выбора* (Яглом).

Современное научное представление об информации очень точно сформулировал **Норберт Винер**, "отец" кибернетики. А именно:

Информация — это обозначение содержания, полученного из внешнего мира в процессе нашего приспособления к нему и приспособления к нему наших чувств.

Люди обмениваются информацией в форме сообщений. Сообщение — это форма представления информации в виде речи, текстов, жестов, взглядов, изображений, цифровых данных, графиков, таблиц и т.п.

Одно и то же информационное сообщение (статья в газете, объявление, письмо, телеграмма, справка, рассказ, чертёж, радиопередача и т.п.) может содержать разное количество информации для разных людей — в зависимости от их предшествующих знаний, от уровня понимания этого сообщения и интереса к нему.

Так, сообщение, составленное на японском языке, не несёт никакой новой информации человеку, не знающему этого языка, но может быть высокоинформативным для человека, владеющего японским. Никакой новой информации не содержит и сообщение, изложенное на знакомом языке, если его содержание непонятно или уже известно.

Информация есть характеристика не сообщения, а соотношения между сообщением и его потребителем. Без наличия потребителя, хотя бы потенциального, говорить об информации бессмысленно.

В случаях, когда говорят **об автоматизированной работе с информацией посредством каких-либо технических устройств**, обычно в первую очередь интересуются не содержанием сообщения, а тем, сколько символов это сообщение содержит.

Применительно к компьютерной обработке данных под информацией понимают некоторую последовательность символических обозначений (букв, цифр, закодированных графических образов и звуков и т.п.), несущую смысловую нагрузку и представленную в понятном компьютеру виде. Каждый новый символ в такой последовательности символов увеличивает информационный объём сообщения.

Более развёрнутое представление о существе рассматриваемых вопросов дается в [11, 41, 42].

1.3. В каком виде существует информация?

Информация может существовать в виде:

- текстов, рисунков, чертежей, фотографий;
- световых или звуковых сигналов;
- радиоволн;
- электрических и нервных импульсов;
- магнитных записей;
- жестов и мимики;
- запахов и вкусовых ощущений;
- хромосом, посредством которых передаются по наследству признаки и свойства организмов и т.д.

Предметы, процессы, явления материального или нематериального свойства, рассматриваемые с точки зрения их информационных свойств, называются информационными объектами.

1.4. Как передаётся информация?

Информация передаётся в форме **сообщений** от некоторого **источника** информации к её **приёмнику** посредством **канала связи** между ними. Источник посылает **передаваемое сообщение**, которое **кодируется в передаваемый сигнал**. Этот сигнал посылается по **каналу связи**. В результате в приёмнике появляется **принимаемый сигнал**, который **декодируется** и становится **принимаемым сообщением**.

канал связи
ИСТОЧНИК -----> ПРИЁМНИК

Примеры:

1. *Сообщение, содержащее информацию о прогнозе погоды, передаётся приёмнику (телезрителю) от источника — специалиста-метеоролога посредством канала связи — телевизионной передающей аппаратуры и телевизора.*
2. *Живое существо своими органами чувств (глаз, ухо, кожа, язык и т.д.) воспринимает информацию из внешнего мира, перерабатывает её в определенную последовательность нервных импульсов, передает импульсы по нервным волокнам, хранит в памяти в виде состояния нейронных структур мозга, воспроизводит в виде звуковых сигналов, движений и т.п., использует в процессе своей жизнедеятельности.*

Передача информации по каналам связи часто сопровождается воздействием **помех**, вызывающих **искажение и потерю информации**.

1.5. Как измеряется количество информации?

Какое количество информации содержится, к примеру, в тексте романа "Война и мир", во фресках Рафаэля или в генетическом коде человека? Ответа на эти вопросы наука не даёт и, по всей вероятности, даст не скоро. **А возможно ли объективно измерить количество информации?** Важнейшим результатом теории информации является следующий вывод:

В определенных, весьма широких условиях можно пренебречь качественными особенностями информации, выразить её количество числом, а также сравнить количество информации, содержащейся в различных группах данных.

В настоящее время получили распространение подходы к определению понятия "количество информации", основанные на том, **что информацию, содержащуюся в сообщении, можно нестрого трактовать в смысле её новизны или, иначе, уменьшения неопределённости наших знаний об объекте.** Эти подходы используют математические понятия *вероятности* и *логарифма*.

Подходы к определению количества информации. Формулы Хартли и Шеннона.

Американский инженер **Р. Хартли** в 1928 г. процесс получения информации рассматривал как выбор одного сообщения из конечного наперёд заданного множества из **N** равновероятных сообщений, а количество информации **I**, содержащееся в выбранном сообщении, определял как двоичный логарифм **N**.

$$\text{Формула Хартли: } I = \log_2 N$$

Допустим, нужно угадать одно число из набора чисел от единицы до ста. По формуле Хартли можно вычислить, какое количество информации для этого требуется: $I = \log_2 100$

> 6,644. Таким образом, сообщение о верно угаданном числе содержит количество информации, приблизительно равное 6,644 единицы информации.

Приведем другие **примеры равновероятных сообщений**:

1. при бросании монеты: "выпала решка", "выпал орел";
2. на странице книги: "количество букв чётное", "количество букв нечётное".

Определим теперь, **являются ли равновероятными сообщения** "первой выйдет из дверей здания женщина" и "первым выйдет из дверей здания мужчина". **Однозначно ответить на этот вопрос нельзя.** Все зависит от того, о каком именно здании идет речь. Если это, например, станция метро, то вероятность выйти из дверей первым одинакова для мужчины и женщины, а если это военная казарма, то для мужчины эта вероятность значительно выше, чем для женщины.

Для задач такого рода американский учёный **Клод Шеннон** предложил в 1948 г. другую **формулу определения количества информации, учитывающую возможную неодинаковую вероятность сообщений в наборе.**

Формула Шеннона: $I = - (p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_N \log_2 p_N)$, где p_i — вероятность того, что именно i -е сообщение выделено в наборе из N сообщений.

Легко заметить, что если вероятности p_1, \dots, p_N равны, то каждая из них равна $1/N$, и формула Шеннона превращается в формулу Хартли.

Помимо двух рассмотренных подходов к определению количества информации, существуют и другие. **Важно помнить, что любые теоретические результаты применимы лишь к определённому кругу случаев, очерченному первоначальными допущениями.**

В качестве единицы информации Клод Шеннон предложил принять один **бит** (англ. *bit* — *binary digit* — двоичная цифра).

Бит в теории информации — количество информации, необходимое для различения двух равновероятных сообщений (типа "орел"—"решка", "чет"—"нечет" и т.п.).

В вычислительной технике битом называют наименьшую "порцию" памяти компьютера, необходимую для хранения одного из двух знаков "0" и "1", используемых для внутримашинного представления данных и команд.

Бит — слишком мелкая единица измерения. На практике чаще применяется более крупная единица — **байт**, равная **восемь битам**. Именно восемь битов требуется для того, чтобы закодировать любой из 256 символов алфавита клавиатуры компьютера ($256=2^8$).

Широко используются также ещё более крупные производные единицы информации:

- **1 Килобайт (Кбайт) = 1024 байт = 210 байт,**
- **1 Мегабайт (Мбайт) = 1024 Кбайт = 220 байт,**
- **1 Гигабайт (Гбайт) = 1024 Мбайт = 230 байт.**

В последнее время в связи с увеличением объёмов обрабатываемой информации входят в употребление такие производные единицы, как:

- **1 Терабайт (Тбайт) = 1024 Гбайт = 240 байт,**
- **1 Петабайт (Пбайт) = 1024 Тбайт = 250 байт.**

За единицу информации можно было бы выбрать количество информации, необходимое для различения, например, десяти равновероятных сообщений. Это будет не двоичная (бит), а десятичная (дит) единица информации.

1.6. Что можно делать с информацией?

Информацию можно:

- | | | |
|-----------------|--------------------|--------------|
| • создавать; | • формализовать; | • собирать; |
| • передавать; | • распространять; | • хранить; |
| • воспринимать; | • преобразовывать; | • искать; |
| • использовать; | • комбинировать; | • измерять; |
| • запоминать; | • обрабатывать; | • разрушать; |
| • принимать; | • делить на части; | • и др. |
| • копировать; | • упрощать; | • |

Все эти процессы, связанные с определенными операциями над информацией, называются **информационными процессами**.

1.7. Какими свойствами обладает информация?

Свойства информации:

- | | |
|---------------------------|-----------------------|
| • достоверность; | • понятность; |
| • полнота; | • доступность; |
| • ценность; | • краткость; |
| • своевременность; | • и др. |

Информация достоверна, если она отражает истинное положение дел. Недостоверная информация может привести к неправильному пониманию или принятию неправильных решений.

Достоверная информация со временем может стать недостоверной, так как она обладает свойством **устаревать**, то есть **перестаёт отражать истинное положение дел**.

Информация полна, если её достаточно для понимания и принятия решений. Как неполная, так и избыточная информация **сдерживает принятие решений или может повлечь ошибки**.

Точность информации определяется степенью ее близости к реальному состоянию объекта, процесса, явления и т.п.

Ценность информации зависит от того, насколько она важна для решения задачи, а также от того, насколько в дальнейшем она найдёт применение в каких-либо видах деятельности человека.

Только **своевременно полученная информация может принести ожидаемую пользу**. Одинаково нежелательны как **преждевременная подача информации** (когда она ещё не может быть усвоена), так и её **задержка**.

Если ценная и своевременная информация выражена непонятным образом, она может стать **бесполезной**.

Информация **становится понятной**, если она выражена языком, на котором говорят те, кому предназначена эта информация.

Информация должна преподноситься в доступной (по уровню восприятия) форме. Поэтому одни и те же вопросы по разному излагаются в школьных учебниках и научных изданиях.

Информацию по одному и тому же вопросу **можно изложить кратко** (сжато, без несущественных деталей) **или пространно** (подробно, многословно). Краткость информации необходима в справочниках, энциклопедиях, учебниках, всевозможных инструкциях.

1.8. Что такое обработка информации?

Обработка информации — получение одних информационных объектов из других информационных объектов путем выполнения некоторых алгоритмов [15].

Обработка является одной из основных операций, выполняемых над информацией, и главным средством увеличения объёма и разнообразия информации.

Средства обработки информации — это всевозможные устройства и системы, созданные человечеством, и в первую очередь, **компьютер** — универсальная машина для обработки информации.

Компьютеры обрабатывают информацию путем выполнения некоторых алгоритмов.

Живые организмы и растения обрабатывают информацию с помощью своих органов и систем.

1.9. Что такое информационные ресурсы и информационные технологии?

Информационные ресурсы — это идеи человечества и указания по их реализации, накопленные в форме, позволяющей их воспроизводство.

Это книги, статьи, патенты, диссертации, научно-исследовательская и опытно-конструкторская документация, технические переводы, данные о передовом производственном опыте и др. [42].

Информационные ресурсы (в отличие от всех других видов ресурсов — трудовых, энергетических, минеральных и т.д.) **тем быстрее растут, чем больше их расходуют.**

Информационная технология — это совокупность методов и устройств, используемых людьми для обработки информации.

Человечество занималось обработкой информации тысячи лет. Первые информационные технологии основывались на использовании счётов и письменности. Около пятидесяти лет назад началось исключительно быстрое развитие этих технологий, что в первую очередь связано с появлением компьютеров.

В настоящее время термин "**информационная технология**" употребляется в связи с **использованием компьютеров для обработки информации.** Информационные технологии охватывают всю **вычислительную технику и технику связи** и, отчасти, — **бытовую электронику, телевидение и радиовещание.**

Они находят применение в промышленности, торговле, управлении, банковской системе, образовании, здравоохранении, медицине и науке, транспорте и связи, сельском хозяйстве, системе социального обеспечения, служат подспорьем людям различных профессий и домохозяйкам.

Народы развитых стран осознают, что **совершенствование информационных технологий представляет самую важную, хотя дорогостоящую и трудную задачу.**

В настоящее время создание крупномасштабных информационно-технологических систем является экономически возможным, и это обуславливает появление национальных исследовательских и образовательных программ, призванных стимулировать их разработку.

1.10. Что понимают под информатизацией общества?

Информатизация общества — организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления организаций, общественных объединений на основе формирования и использования информационных ресурсов [53].

Цель информатизации — улучшение качества жизни людей за счет увеличения производительности и облегчения условий их труда.

Информатизация — это сложный социальный процесс, связанный со значительными изменениями в образе жизни населения. Он требует серьёзных усилий на многих направлениях, включая ликвидацию компьютерной неграмотности, формирование культуры использования новых информационных технологий и др.

1.11. Вопросы для самоконтроля

- 1.1. Что означает термин "информатика" и каково его происхождение?
- 1.2. Какие области знаний и административно-хозяйственной деятельности официально закреплены за понятием "информатика" с 1978 года?
- 1.3. Какие сферы человеческой деятельности и в какой степени затрагивает информатика?
- 1.4. Назовите основные составные части информатики и основные направления её применения.
- 1.5. Что подразумевается под понятием "информация" в бытовом, естественно-научном и техническом смыслах?
- 1.6. Приведите примеры знания фактов и знания правил. Назовите новые факты и новые правила, которые Вы узнали за сегодняшний день.
- 1.7. От кого (или чего) человек принимает информацию? Кому передает информацию?
- 1.8. Где и как человек хранит информацию?
- 1.9. Что необходимо добавить в систему "источник информации — приёмник информации", чтобы осуществлять передачу сообщений?
- 1.10. Какие типы действий выполняет человек с информацией?
- 1.11. Приведите примеры ситуаций, в которых информация
 - а) создаётся;
 - б) обрабатывается;
 - в) запоминается;
 - д) копируется;
 - е) воспринимается;
 - ж) измеряется;
 - и) передаётся;
 - к) разрушается;
 - л) ищется;

г) делится на части;

з) принимается;

м) упрощается.

1.12. Приведите примеры обработки информации человеком. Что является результатами этой обработки?

1.13. Приведите примеры информации:

- а) достоверной и недостоверной;
- б) полной и неполной;
- в) ценной и малоценной;
- г) своевременной и несвоевременной;
- д) понятной и непонятной;
- е) доступной и недоступной для усвоения;
- ж) краткой и пространной.

1.14. Назовите системы сбора и обработки информации в теле человека.

1.15. Приведите примеры технических устройств и систем, предназначенных для сбора и обработки информации.

1.16. От чего зависит информативность сообщения, принимаемого человеком?

1.17. Почему количество информации в сообщении удобнее оценивать не по степени увеличения знания об объекте, а по степени уменьшения неопределённости наших знаний о нём?

1.18. Как определяется единица измерения количества информации?

1.19. В каких случаях и по какой формуле можно вычислить количество информации, содержащейся в сообщении?

1.20. Почему в формуле Хартли за основание логарифма взято число 2?

1.21. При каком условии формула Шеннона переходит в формулу Хартли?

1.22. Что определяет термин "бит" в теории информации и в вычислительной технике?

1.23. Приведите примеры сообщений, информативность которых можно однозначно определить.

1.24. Приведите примеры сообщений, содержащих один (два, три) бит информации.

1.12. Упражнения

1.1. Запишите множество вариантов загорания двух светофоров, расположенных на соседних перекрёстках.

1.2. Три человека, Иванов, Петров и Сидоров, образуют очередь. Запишите все возможные варианты образования этой очереди.

1.3. Назовите все возможные комбинации из двух различных нот (всего нот семь: до, ре, ми, фа, соль, ля, си).

1.4. Пусть голосуют 3 человека (голосование "да"/"нет"). Запишите все возможные исходы голосования.

1.5. Предположим, что имеются 3 автомобильные дороги, идущие от Парижа до Тулузы, и 4 — от Тулузы до Мадрида. Сколькими способами можно выбрать дорогу от Парижа в Мадрид через Тулузу? Попробуйте найти систематический метод для последовательного нахождения решения так, чтобы можно было составить список способов, не пропустив ни одного из них.

1.6. Поезд находится на одном из восьми путей. Сколько бит информации содержит сообщение о том, где находится поезд?

1.7. Сколько существует различных двоичных последовательностей из одного, двух, трех, четырёх, восьми символов?

1.8. Каков информационный объём сообщения *"Я помню чудное мгновенье"* при условии, что один символ кодируется одним байтом и соседние слова разделены одним пробелом?

1.9. Определите приблизительно информационный объём:

- а) этой страницы книги;
- б) всей книги;
- в) поздравительной открытки.

1.10. Сколько бит необходимо, чтобы закодировать оценки: "неудовлетворительно", "удовлетворительно", "хорошо" и "отлично"?

1.11. Сколько различных символов, закодированных байтами, содержится в сообщении: 1101001100011100110100110001110001010111 ?

1.12. Сколько байт памяти необходимо, чтобы закодировать изображение на экране компьютерного монитора, который может отображать 1280 точек по горизонтали и 1024 точек по вертикали при 256 цветах?

1.13. Решите уравнение: 8^x (бит) = 32 (Кбайт).

1.14. Решите систему уравнений

$$\begin{cases} 2^{x+2} \text{ (бит)} = 8^{y-5} \text{ (Кбайт)}, \\ 2^{2y-1} \text{ (Мбайт)} = 16^{x-3} \text{ (бит)}. \end{cases}$$

1.15. Определите правила формирования приведённых ниже последовательностей и вставьте пропущенные числа:

- | | | |
|------------------------|-----------------------------|-----------------|
| а) 1, 3, 5, ..., 9; | ж) 128, 64, 32, ..., 8; | н) 15 (27) 42 |
| б) 20, 15, ..., 5; | з) 4, 9, 17, 35, ..., 139; | 30 (...) 55; |
| в) 1, 2, 4, ..., 16; | и) 1, 2, 2, 4, 8, ..., 256; | о) 10 (50) 15 |
| г) 1, 4, 9, ..., 25; | к) 2, 3, 10, 15, ..., 35; | 17 (...) 20; |
| д) 1, 8, 27, ..., 125; | л) 1, 3, 3, 9, ..., 6561; | п) 143 (56) 255 |
| е) 1, 2, 6, ..., 120; | м) к, о, ж, з, г, ..., ф; | 218 (...) 114. |

Ответы — Раздел 1. Введение в информатику

1.1. Обозначим номера светофоров цифрами 1 и 2, а цвета их загорания — буквами З (зелёный), Ж (жёлтый) и К (красный). Тогда искомое множество содержит следующие 9 элементов: $З_1 - З_2$, $З_1 - Ж_2$, $З_1 - К_2$, $Ж_1 - З_2$, $Ж_1 - Ж_2$, $Ж_1 - К_2$, $К_1 - З_2$, $К_1 - Ж_2$, $К_1 - К_2$.

1.2. Вариантов всего шесть:

- | | |
|-----------------------------|-----------------------------|
| 1) Иванов, Петров, Сидоров; | 4) Петров, Сидоров, Иванов; |
| 2) Иванов, Сидоров, Петров; | 5) Сидоров, Иванов, Петров; |
| 3) Петров, Иванов, Сидоров; | 6) Сидоров, Петров, Иванов. |
-

1.3. Комбинаций всего 42:
"до-ре", "до-ми", ..., "до-си", "ре-до", "ре-ми", ..., "ре-си", "ми-до", "ми-ре", ..., "ми-си",
..., "си-до", "си-ре", ..., "си-ля".

1.4. Исходов всего восемь:
"да-да-да", "да-да-нет", "да-нет-да", "да-нет-нет", "нет-да-да", "нет-да-нет", "нет-нет-да",
"нет-нет-нет".

1.5. Двенадцатью способами.

1.6. 3 бита информации ($8 = 2^3$).

1.7. Двоичных последовательностей из одного бита всего $2^1 = 2$, из двух битов — $2^2 = 4$, из трех битов — $2^3 = 8$, из четырех битов — $2^4 = 16$, из восьми битов — $2^8 = 256$.

1.8. 24 байта, или 192 бита.

1.10 Два бита: 00 — "неуд.", 01 — "удов.", 10 — "хор.", 11 — "отл."

1.11. Разбиваем сообщение на восьмёрки битов (то есть, на байты):

01001100 01110011 01001100 01110011 01010111.

Сравнивая байты между собой, видим, что первый и третий, а также второй и четвёртый байты одинаковые. Следовательно, различных символов всего три.

1.12. Всего на экране монитора $1280 \times 1024 = 1310720$ точек. Для кодирования каждой из точек, которые могут быть окрашены в 256 цветов ($256 = 2^8$) требуется 8 бит или 1 байт. Т.о., для кодирования всего изображения требуется 1310720 байт $= 1,25 \cdot 2^{20}$ байт $= 1,25$ Мбайт.

1.13. Выравниваем размерности в левой и правой частях уравнения с учётом того, что 1 Кбайт $= 2^{13}$ бит. Затем приводим обе части к одному основанию 2. Имеем: $2^{3x} = 2^5 \cdot 2^{13}$ или $2^{3x} = 2^{18}$. Переходим к равносильному уравнению $3x = 18$, откуда $x = 18:3 = 6$.

1.14. Выравниваем размерности с учётом того, что 1 Кбайт $= 2^{13}$ бит и 1 Мбайт $= 2^{23}$ бит. Приводим оба уравнения к одному основанию 2.

$$2^{x+2} = 2^{3(y-5)} \cdot 2^{13},$$

$$x+2 = 3y-15+13,$$

$$2^{2y-1} \cdot 2^{23} = 2^{4(x-3)}$$

$$2y-1+23 = 4x-12,$$

откуда $x = 11$, $y = 5$.

1.15.

- а)** 7 (чтобы получить следующее число, нужно к предыдущему прибавить 2: $a_1 = 1$, $a_i = a_{i-1} + 2$, $i = 2, 3, \dots$);
- б)** 10 (чтобы получить следующее число, нужно от предыдущего отнять 5: $a_1 = 20$, $a_i = a_{i-1} - 5$, $i = 2, 3, \dots$);
- в)** 8 (чтобы получить следующее число, нужно предыдущее умножить на 2: $a_1 = 1$, $a_i = a_{i-1} \cdot 2$, $i = 2, 3, \dots$);
- г)** 16 (возвести в квадрат числа 1, 2, 3, ... : $a_i = i^2$, $i = 1, 2, 3, \dots$);
- д)** 64 (возвести в куб числа 1, 2, 3, ... : $a_i = i^3$, $i = 1, 2, 3, \dots$);
- е)** 24 (чтобы получить очередное число, нужно предыдущее умножить на номер числа: $a_1 = 1$, $a_i = a_{i-1} \cdot i$, $i = 2, 3, \dots$);
- ж)** 8 (чтобы получить следующее число, нужно предыдущее разделить на 2: $a_1 = 128$, $a_i = a_{i-1} : 2$, $i = 2, 3, \dots$);
- з)** 69 (чтобы получить следующее число, нужно предыдущее умножить на 2 и к полученному произведению поочередно прибавлять и вычитать единицу: $a_1 = 4$, $a_i = 2a_{i-1} + (-1)^i$, $i = 2, 3, \dots$);
- и)** 32 ($a_1 = 1$, $a_2 = 2$, $a_i = a_{i-1} \cdot a_{i-2}$, $i = 3, 4, \dots$);
- к)** 26 ($a_1 = 2$, $a_i = i^2 + (-1)^{i-1}$, $i = 2, 3, \dots$);
- л)** 81 ($a_1 = 1$, $a_2 = 3$, $a_i = a_1 \cdot a_2 \cdot \dots \cdot a_{i-1}$, $i = 3, 4, \dots$);
- м)** с (выписаны первые буквы цветов радуги: с — "синий");
- н)** 25 (число в скобках есть разность между числами вне скобок);
- о)** 74 (удвоенная сумма чисел, стоящих вне скобок);
- п)** 52 (полуразность чисел, стоящих вне скобок).
-

Глава 2. Общие принципы организации и работы компьютеров

2.1. Что такое компьютер?

Компьютер (англ. computer — вычислитель) представляет собой программируемое электронное устройство, способное обрабатывать данные и производить вычисления, а также выполнять другие задачи манипулирования символами [51].

Существует два основных класса компьютеров:

- **цифровые компьютеры**, обрабатывающие данные в виде двоичных кодов;
- **аналоговые компьютеры**, обрабатывающие непрерывно меняющиеся физические величины (электрическое напряжение, время и т.д.), которые являются аналогами вычисляемых величин.

Поскольку в настоящее время подавляющее большинство компьютеров являются цифровыми, далее будем рассматривать только этот класс компьютеров и слово "компьютер" употреблять в значении "*цифровой компьютер*".

Основу компьютеров образует аппаратура (HardWare), построенная, в основном, с использованием электронных и электромеханических элементов и устройств. Принцип действия компьютеров состоит в выполнении программ (SoftWare) — заранее заданных, четко определённых последовательностей арифметических, логических и других операций.

Любая компьютерная программа представляет собой последовательность отдельных команд.

Команда — это описание операции, которую должен выполнить компьютер. Как правило, у команды есть свой *код* (условное обозначение), *исходные данные* (операнды) и *результат*.

Например, у команды "*сложить два числа*" операндами являются слагаемые, а результатом — их сумма. А у команды "*стоп*" операндов нет, а результатом является прекращение работы программы.

Результат команды вырабатывается по точно определенным для данной команды правилам, заложенным в конструкцию компьютера.

Совокупность команд, выполняемых данным компьютером, называется системой команд этого компьютера.

Компьютеры работают с очень высокой скоростью, составляющей миллионы — сотни миллионов операций в секунду.

2.2. Как устроен компьютер?

Разнообразие современных компьютеров очень велико. Но их структуры основаны на **общих логических принципах**, позволяющих выделить в любом компьютере следующие **главные устройства**:

- **память** (запоминающее устройство, ЗУ), состоящую из перенумерованных ячеек;
- **процессор**, включающий в себя **устройство управления (УУ)** и **арифметико-логическое устройство (АЛУ)**;
- **устройство ввода**;
- **устройство вывода**.

Эти устройства соединены **каналами связи**, по которым передается информация.

Основные устройства компьютера и связи между ними представлены на схеме (рис. 2.1). Жирными стрелками показаны пути и направления движения информации, а простыми стрелками — пути и направления передачи управляющих сигналов.

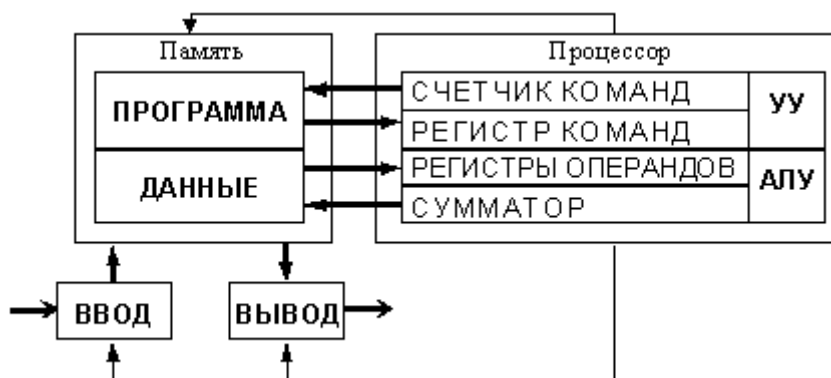


Рис. 2.1. Общая схема компьютера

Функции памяти:

- **приём информации** из других устройств;
- **запоминание информации**;
- **выдача информации** по запросу в другие устройства машины.

Функции процессора:

- **обработка данных по заданной программе** путем выполнения арифметических и логических операций;
- **программное управление работой устройств** компьютера.

Та часть процессора, которая выполняет команды, называется **арифметико-логическим устройством (АЛУ)**, а другая его часть, выполняющая функции управления устройствами, называется **устройством управления (УУ)**.

Обычно эти два устройства выделяются чисто условно, **конструктивно они не разделены.**

В составе процессора имеется ряд специализированных дополнительных ячеек памяти, называемых **регистрами.**

Регистр выполняет функцию кратковременного хранения числа или команды. Над содержимым некоторых регистров специальные электронные схемы могут выполнять некоторые манипуляции. Например, "вырезать" отдельные части команды для последующего их использования или выполнять определенные арифметические операции над числами.

Основным элементом регистра является электронная схема, называемая **триггером**, которая способна хранить одну двоичную цифру (разряд двоичного кода). Логическая схема триггера описана в разделе 5.7.

Регистр представляет собой совокупность триггеров, связанных друг с другом определенным образом общей системой управления.

Существует несколько типов регистров, отличающихся видом выполняемых операций. Некоторые важные регистры имеют свои названия, например:

- **сумматор** — регистр АЛУ, участвующий в выполнении каждой операции;
- **счетчик команд** — регистр УУ, содержимое которого соответствует адресу очередной выполняемой команды; служит для автоматической выборки программы из последовательных ячеек памяти;
- **регистр команд** — регистр УУ для хранения кода команды на период времени, необходимый для ее выполнения. Часть его разрядов используется для хранения кода операции, остальные — для хранения кодов адресов операндов.

2.3. На каких принципах построены компьютеры?

В основу построения подавляющего большинства компьютеров положены следующие общие принципы, сформулированные в 1945 г. американским ученым **Джоном фон Нейманом**.



Рис. 2.2. Джон фон Нейман, 1945 г.

1. Принцип программного управления. Из него следует, что программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

Выборка программы из памяти осуществляется с помощью счетчика команд. Этот регистр процессора последовательно увеличивает хранимый в нем адрес очередной команды на длину команды.

А так как команды программы расположены в памяти друг за другом, то тем самым организуется выборка цепочки команд из последовательно расположенных ячеек памяти.

Если же нужно после выполнения команды перейти не к следующей, а к какой-то другой, используются команды условного или безусловного переходов, которые заносят в счетчик команд номер ячейки памяти, содержащей следующую команду. Выборка команд из памяти прекращается после достижения и выполнения команды “*stop*”.

Таким образом, процессор исполняет программу автоматически, без вмешательства человека.

2. Принцип однородности памяти. Программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными. Это открывает целый ряд возможностей. Например, программа в процессе своего выполнения также может подвергаться переработке, что позволяет задавать в самой программе правила получения некоторых ее частей (так в программе организуется выполнение циклов и подпрограмм). Более того, команды одной программы могут быть получены как результаты исполнения другой программы. На этом принципе основаны методы трансляции — перевода текста программы с языка программирования высокого уровня на язык конкретной машины.

3. Принцип адресности. Структурно основная память состоит из перенумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Отсюда следует возможность давать имена областям памяти, так, чтобы к

запомненным в них значениям можно было впоследствии обращаться или менять их в процессе выполнения программ с использованием присвоенных имен.

Компьютеры, построенные на этих принципах, относятся к типу **фон-неймановских**. Но существуют компьютеры, принципиально отличающиеся от фон-неймановских. Для них, например, может **не выполняться принцип программного управления**, т.е. они могут работать без “счетчика команд”, указывающего текущую выполняемую команду программы. Для обращения к какой-либо переменной, хранящейся в памяти, этим компьютерам **не обязательно давать ей имя**. Такие компьютеры называются **не-фон-неймановскими**.

2.4. Что такое команда?

Команда — это описание элементарной операции, которую должен выполнить компьютер.

В общем случае, команда содержит следующую информацию:

- код выполняемой операции;
- указания по определению **операндов** (или их адресов);
- указания по размещению получаемого **результата**.

В зависимости от количества операндов, команды бывают:

- одноадресные;
- двухадресные;
- трехадресные;
- переменнаяадресные.

Команды хранятся в ячейках памяти в двоичном коде.

В современных компьютерах **длина команд переменная** (обычно от двух до четырех байтов), **а способы указания адресов переменных весьма разнообразны**. В адресной части команды может быть указан, например:

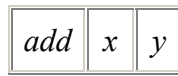
- сам операнд (число или символ);
- адрес операнда (номер байта, начиная с которого расположен операнд);
- адрес адреса операнда (номер байта, начиная с которого расположен адрес операнда), и др.

Рассмотрим несколько **возможных вариантов команды сложения** (англ. add — сложение), при этом вместо цифровых кодов и адресов будем пользоваться условными обозначениями:

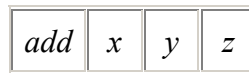
- **одноадресная команда *add x*** (содержимое ячейки *x* сложить с содержимым сумматора, а результат оставить в сумматоре)

<i>add</i>	<i>x</i>
------------	----------

-
- **двухадресная команда *add x, y*** (сложить содержимое ячеек *x* и *y*, а результат поместить в ячейку *y*)



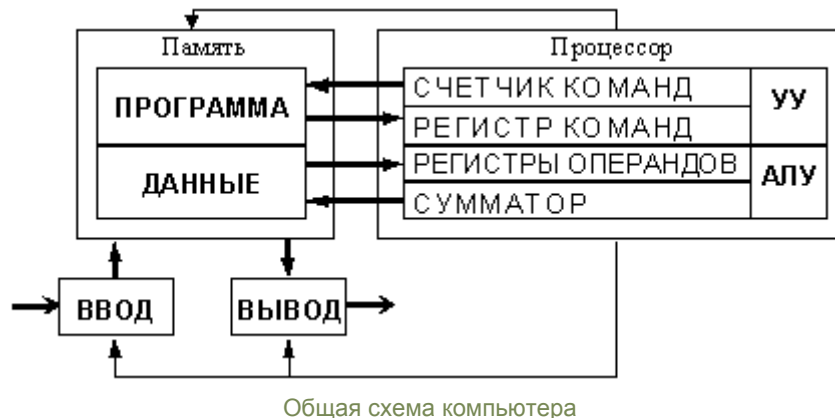
-
- **трехадресная команда *add x, y, z*** (содержимое ячейки *x* сложить с содержимым ячейки *y*, сумму поместить в ячейку *z*)



-

2.5. Как выполняется команда?

Выполнение команды можно проследить по схеме:



Как правило, этот процесс разбивается на следующие этапы:

- из ячейки памяти, адрес которой хранится в счетчике команд, выбирается очередная команда; содержимое счетчика команд при этом увеличивается на длину команды;
- выбранная команда передается в устройство управления на регистр команд;
- устройство управления расшифровывает адресное поле команды;
- по сигналам УУ операнды считываются из памяти и записываются в АЛУ на специальные регистры операндов;
- УУ расшифровывает код операции и выдает в АЛУ сигнал выполнить соответствующую операцию над данными;
- результат операции либо остается в процессоре, либо отправляется в память, если в команде был указан адрес результата;
- все предыдущие этапы повторяются до достижения команды “стоп”.

2.6. Что такое архитектура и структура компьютера?

При рассмотрении компьютерных устройств принято различать их архитектуру и структуру.

Архитектурой компьютера называется его описание на некотором общем уровне, включающее описание пользовательских возможностей программирования, системы команд, системы адресации, организации памяти и т.д. Архитектура определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: процессора, оперативного ЗУ, внешних ЗУ и периферийных устройств. Общность архитектуры разных компьютеров обеспечивает их совместимость с точки зрения пользователя.

Структура компьютера — это совокупность его функциональных элементов и связей между ними. Элементами могут быть самые различные устройства — от основных логических узлов компьютера до простейших схем. Структура компьютера графически представляется в виде структурных схем, с помощью которых можно дать описание компьютера на любом уровне детализации.

Наиболее распространены следующие архитектурные решения.

- **Классическая архитектура** (архитектура фон Неймана) — одно арифметико-логическое устройство (АЛУ), через которое проходит поток данных, и одно устройство управления (УУ), через которое проходит поток команд — программа (рис. 2.1). Это **однопроцессорный компьютер**. К этому типу архитектуры относится и архитектура персонального компьютера с **общей шиной**, подробно рассмотренная в разделе 2.18 (рис. 2.26). Все функциональные блоки здесь связаны между собой общей шиной, называемой также **системной магистралью**.

Физически магистраль представляет собой многопроводную линию с гнездами для подключения электронных схем. Совокупность проводов магистрали разделяется на отдельные группы: шину адреса, шину данных и шину управления.

Периферийные устройства (принтер и др.) подключаются к аппаратуре компьютера через специальные **контроллеры** — **устройства управления периферийными устройствами**.

Контроллер — устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

- **Многопроцессорная архитектура**. Наличие в компьютере нескольких процессоров означает, что **параллельно может быть организовано много потоков данных и много потоков команд**. Таким образом, параллельно могут выполняться несколько фрагментов одной задачи. Структура такой машины, имеющей общую оперативную память и несколько процессоров, представлена на рис. 2.3.

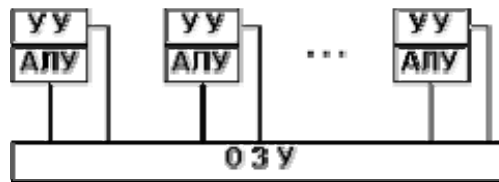


Рис. 2.3. Архитектура многопроцессорного компьютера

● **Многомашинная вычислительная система.** Здесь несколько процессоров, входящих в вычислительную систему, не имеют общей оперативной памяти, а имеют каждый свою (локальную). Каждый компьютер в многомашинной системе имеет классическую архитектуру, и такая система применяется достаточно широко. Однако эффект от применения такой вычислительной системы может быть получен только при решении задач, имеющих очень специальную структуру: она должна разбиваться на столько слабо связанных подзадач, сколько компьютеров в системе.

Преимущество в быстродействии многопроцессорных и многомашинных вычислительных систем перед однопроцессорными очевидно.

● **Архитектура с параллельными процессорами.** Здесь несколько АЛУ работают под управлением одного УУ. Это означает, что множество данных может обрабатываться по одной программе — то есть по одному потоку команд. Высокое быстродействие такой архитектуры можно получить только на задачах, в которых одинаковые вычислительные операции выполняются одновременно на различных однотипных наборах данных. Структура таких компьютеров представлена на рис. 2.4.

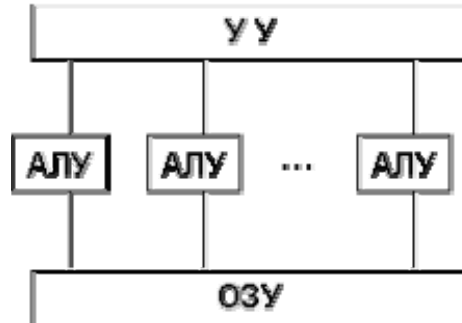


Рис. 2.4. Архитектура с параллельным процессором

В современных машинах часто присутствуют элементы различных типов архитектурных решений. Существуют и такие архитектурные решения, которые радикально отличаются от рассмотренных выше.

2.7. Что такое центральный процессор?

Центральный процессор (CPU, от англ. Central Processing Unit) — это основной рабочий компонент компьютера, который выполняет арифметические и логические операции, заданные программой, управляет вычислительным процессом и координирует работу всех устройств компьютера.

Центральный процессор в общем случае содержит в себе:

- арифметико-логическое устройство;
- шины данных и шины адресов;
- регистры;
- счетчики команд;
- кэш — очень быструю память малого объема (от 8 до 512 Кбайт);
- математический сопроцессор чисел с плавающей точкой.

Современные процессоры выполняются в виде **микروпроцессоров**. Физически микروпроцессор представляет собой интегральную схему — тонкую пластинку кристаллического кремния прямоугольной формы площадью всего несколько квадратных миллиметров, на которой размещены схемы, реализующие все функции процессора. Кристалл-пластинка обычно помещается в пластмассовый или керамический плоский корпус и соединяется золотыми проводками с металлическими штырьками, чтобы его можно было присоединить к системной плате компьютера.

Микропроцессор **Intel Pentium 4** — наиболее совершенный и мощный процессор выпуска 2001 г. с тактовой частотой до 2 Гигагерц, представлен на рисунке 2.5 примерно в натуральную величину. Он предназначен для работы приложений, требующих высокой производительности процессора, таких, как передача видео и звука по Интернет, создание видео-материалов, распознавание речи, обработка трехмерной графики, игры.

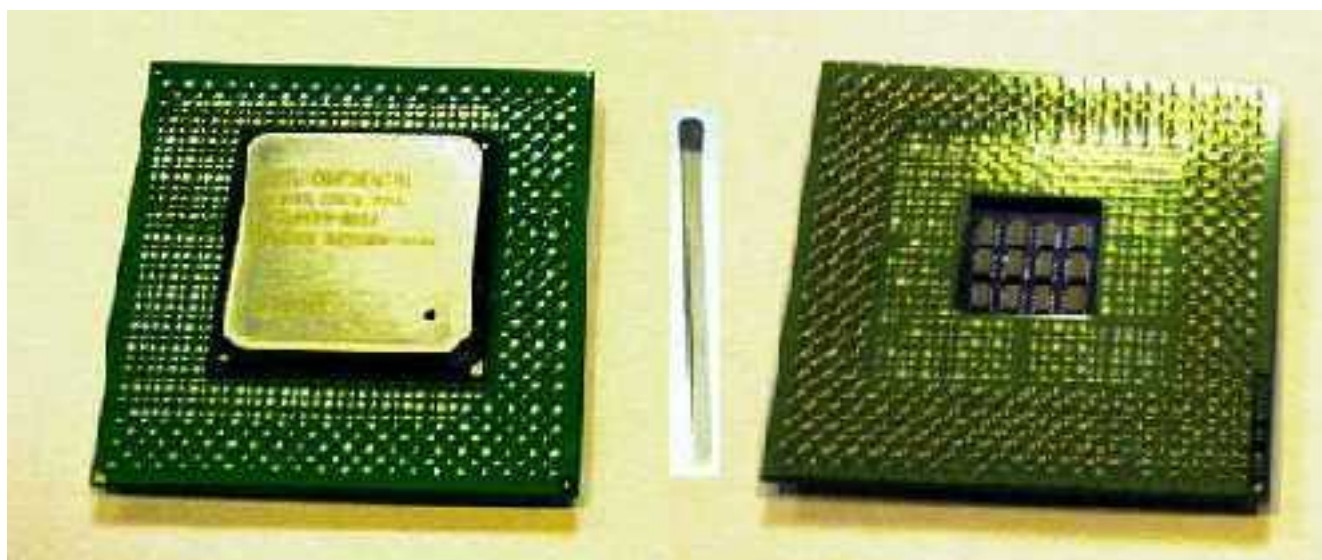


Рис. 2.5. Микропроцессор Pentium 4. Вид сверху (слева) и вид снизу (справа)

В вычислительной системе может быть несколько параллельно работающих процессоров; такие системы называются **многопроцессорными**.

2.8. Как устроена память?

Память компьютера построена из двоичных запоминающих элементов — **битов**, объединенных в группы по 8 битов, которые называются **байтами**. (Единицы измерения памяти совпадают с единицами измерения информации). Все байты пронумерованы. Номер байта называется его **адресом**.

Байты могут объединяться в ячейки, которые называются также **словами**. Для каждого компьютера характерна определенная длина слова — два, четыре или восемь байтов. Это не исключает использования ячеек памяти другой длины (например,

полуслово, двойное слово). Как правило, в одном машинном слове может быть представлено либо одно целое число, либо одна команда. Однако, допускаются переменные форматы представления информации. Разбиение памяти на слова для четырехбайтовых компьютеров представлено в таблице:

Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
ПОЛУСЛОВО		ПОЛУСЛОВО		ПОЛУСЛОВО		ПОЛУСЛОВО	
СЛОВО				СЛОВО			
ДВОЙНОЕ СЛОВО							

Широко используются и более крупные [производные единицы](#) объема памяти: **Килобайт, Мегабайт, Гигабайт**, а также, в последнее время, **Терабайт** и **Петабайт**.

Современные компьютеры имеют много разнообразных запоминающих устройств, которые сильно отличаются между собой по назначению, временным характеристикам, объёму хранимой информации и стоимости хранения одинакового объёма информации. Различают два основных вида памяти — [внутреннюю](#) и [внешнюю](#).

2.9. Какие устройства образуют внутреннюю память?

В состав внутренней памяти входят **оперативная память, кэш-память** и **специальная память**.

1. Оперативная память

Оперативная память (ОЗУ, англ. RAM, Random Access Memory — память с произвольным доступом) — это быстрое запоминающее устройство не очень большого объёма, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

Оперативная память используется только для **временного хранения данных и программ**, так как, **когда машина выключается, все, что находилось в ОЗУ, пропадает**. Доступ к элементам оперативной памяти **прямой** — это означает, что **каждый байт памяти имеет свой индивидуальный адрес**.

Объём ОЗУ обычно составляет **от 32 до 512 Мбайт**. Для несложных административных задач бывает достаточно и 32 Мбайт ОЗУ, но сложные задачи компьютерного дизайна могут потребовать от 512 Мбайт до 2 Гбайт ОЗУ.

Обычно ОЗУ **исполняется из интегральных микросхем памяти SDRAM** (синхронное динамическое ОЗУ). Каждый информационный бит в SDRAM запоминается в виде электрического заряда крохотного конденсатора, образованного в структуре полупроводникового кристалла. Из-за токов утечки такие конденсаторы быстро разряжаются, и их периодически (примерно каждые 2 миллисекунды) подзаряжают специальные устройства. Этот процесс называется **регенерацией памяти** (Refresh Memory). Микросхемы SDRAM имеют **ёмкость 16 — 256 Мбит** и более. Они устанавливаются в корпуса и собираются в **модули памяти**.

Большинство современных компьютеров комплектуются модулями типа DIMM (Dual-In-line Memory Module — модуль памяти с двухрядным расположением микросхем). В компьютерных системах на самых современных процессорах используются высокоскоростные модули Rambus DRAM (RIMM) и DDR DRAM.



Рис. 2.6. Микросхемы памяти RIMM (сверху) и DIMM (снизу)

Модули памяти характеризуются такими параметрами, как **объем** — (16, 32, 64, 128, 256 или 512 Мбайт), **число микросхем**, **паспортная частота** (100 или 133 МГц), **время доступа к данным** (6 или 7 наносекунд) и **число контактов** (72, 168 или 184). В 2001 г. начинается выпуск модулей памяти на **1 Гбайт** и опытных образцов модулей на **2 Гбайта**.

2. Кэш-память

Кэш (англ. cache), или **сверхоперативная память** — очень быстрое ЗУ небольшого объема, которое используется при обмене данными между микропроцессором и оперативной памятью для компенсации разницы в скорости обработки информации процессором и несколько менее быстродействующей оперативной памятью.

Кэш-памятью управляет специальное устройство — **контроллер**, который, анализируя выполняемую программу, пытается **предвидеть, какие данные и команды вероятнее всего понадобятся в ближайшее время процессору, и подкачивает их в кэш-память**. При этом возможны как "**попадания**", так и "**промахи**". В случае попадания, то есть, если в кэш подкачаны нужные данные, извлечение их из памяти происходит без задержки. Если же требуемая информация в кэше отсутствует, то процессор считывает её непосредственно из оперативной памяти. Соотношение числа попаданий и промахов определяет эффективность кэширования.

Кэш-память реализуется на **микросхемах статической памяти SRAM** (Static RAM), более быстродействующих, дорогих и малоёмких, чем DRAM (SDRAM). Современные микропроцессоры имеют **встроенную кэш-память**, так называемый **кэш первого уровня** размером 8, 16 или 32 Кбайт. Кроме того, на системной плате компьютера может быть установлен **кэш второго уровня** ёмкостью 256, 512 Кбайт и выше.

3. Специальная память

К устройствам специальной памяти относятся **постоянная память (ROM)**, **перепрограммируемая постоянная память (Flash Memory)**, **память CMOS RAM**, питаемая от батарейки, **видеопамять** и некоторые другие виды памяти.

Постоянная память (ПЗУ, англ. ROM, Read Only Memory — память только для чтения) — энергонезависимая память, используется для хранения данных, которые никогда не потребуют изменения. Содержание памяти специальным образом "**зашивается**" в устройстве при его изготовлении для постоянного хранения. Из ПЗУ можно только

читать.

Перепрограммируемая постоянная память (Flash Memory) — энергонезависимая память, допускающая многократную перезапись своего содержимого с дискеты.

Прежде всего в постоянную память записывают программу управления работой самого процессора. В ПЗУ находятся программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.

Важнейшая микросхема постоянной или Flash-памяти — модуль BIOS. Роль BIOS двоякая: с одной стороны это неотъемлемый элемент аппаратуры, а с другой стороны — важный модуль любой операционной системы.

BIOS (Basic Input/Output System — базовая система ввода-вывода) — совокупность программ, предназначенных для автоматического тестирования устройств после включения питания компьютера и загрузки операционной системы в оперативную память.

Разновидность постоянного ЗУ — **CMOS RAM**.

CMOS RAM — это память с невысоким быстродействием и минимальным энергопотреблением от батарейки. Используется для хранения информации о конфигурации и составе оборудования компьютера, а также о режимах его работы.



Интегральные схемы BIOS и CMOS

Содержимое CMOS изменяется специальной программой **Setup**, находящейся в BIOS (англ. Set-up — устанавливать, читается "сетап").

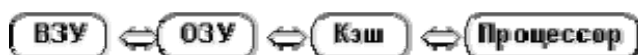
Для хранения графической информации используется **видеопамять**.

Видеопамять (VRAM) — разновидность оперативного ЗУ, в котором хранятся закодированные изображения. Это ЗУ организовано так, что его содержимое доступно

сразу двум устройствам — процессору и дисплею. Поэтому изображение на экране меняется одновременно с обновлением видеоданных в памяти.

2.10. Какие устройства образуют внешнюю память?

Внешняя память (ВЗУ) предназначена для длительного хранения программ и данных, и целостность её содержимого не зависит от того, включен или выключен компьютер. В отличие от оперативной памяти, **внешняя память не имеет прямой связи с процессором**. Информация от ВЗУ к процессору и наоборот циркулирует примерно по следующей цепочке:

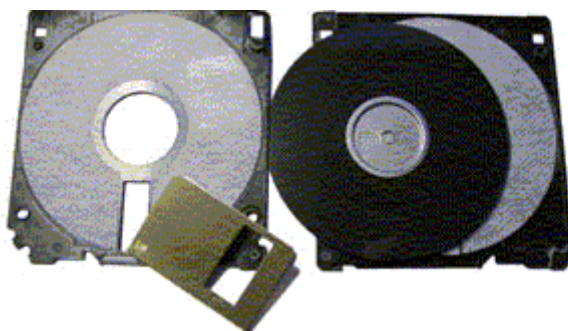


В состав внешней памяти компьютера входят:

- накопители на **жёстких магнитных дисках**;
- накопители на **гибких магнитных дисках**;
- накопители на **компакт-дисках**;
- накопители на **магнито-оптических компакт-дисках**;
- накопители на **магнитной ленте** (стримеры) и др.

1. Накопители на гибких магнитных дисках

Гибкий диск (англ. *floppy disk*), или лискета, — носитель небольшого объема информации, представляющий собой гибкий пластиковый диск в защитной оболочке. Используется для переноса данных с одного компьютера на другой и для распространения программного обеспечения.



Устройство дискеты

Дискета состоит из круглой полимерной подложки, покрытой с обеих сторон магнитным окислом и помещенной в пластиковую упаковку, на внутреннюю поверхность которой нанесено очищающее покрытие. В упаковке сделаны с двух сторон радиальные прорезы, через которые головки считывания/записи накопителя получают доступ к диску.

Способ записи двоичной информации на магнитной среде называется **магнитным кодированием**. Он заключается в том, что магнитные домены в среде выстраиваются вдоль дорожек в направлении приложенного магнитного поля своими северными и

южными полюсами. Обычно устанавливается однозначное соответствие между двоичной информацией и ориентацией магнитных доменов.

Информация записывается по концентрическим *дорожкам* (*трекам*), которые делятся на *секторы*. Количество дорожек и секторов зависит от типа и формата дискеты. Сектор хранит минимальную порцию информации, которая может быть записана на диск или считана. Ёмкость сектора постоянна и составляет 512 байтов.



Рис. 2.7. Поверхность магнитного диска

В настоящее время наибольшее распространение получили **дискеты со следующими характеристиками**: диаметр 3,5 дюйма (89 мм), ёмкость 1,44 Мбайт, число дорожек 80, количество секторов на дорожках 18.

Дискета устанавливается в **накопитель на гибких магнитных дисках** (англ. **floppy-disk drive**), **автоматически в нем фиксируется**, после чего механизм накопителя раскручивается до частоты вращения 360 мин^{-1} . В накопителе вращается сама дискета, магнитные головки остаются неподвижными. Дискета вращается только при обращении к ней. Накопитель связан с процессором через **контроллер гибких дисков**.

В последнее время появились трехдюймовые дискеты, которые могут хранить **до 3 Гбайт** информации. Они изготавливаются по новой технологии **Nano2** и требуют специального оборудования для чтения и записи.

2. Накопители на жестких магнитных дисках

Если гибкие диски — это средство переноса данных между компьютерами, то **жесткий диск — информационный склад компьютера**.

Накопитель на жестких магнитных дисках (англ. HDD — Hard Disk Drive) или **винчестерский накопитель** — это наиболее массовое запоминающее устройство большой ёмкости, в котором носителями информации являются круглые алюминиевые пластины — **платтеры**, обе поверхности которых покрыты слоем магнитного материала. Используется для постоянного хранения информации — программ и данных.



Рис. 2.8. Винчестерский накопитель со снятой крышкой корпуса

Как и у дискеты, рабочие поверхности платтеров разделены на кольцевые концентрические дорожки, а дорожки — на секторы. Головки считывания-записи вместе с их несущей конструкцией и дисками заключены в герметически закрытый корпус, называемый **модулем данных**. При установке модуля данных на дисковод он автоматически соединяется с системой, подкачивающей очищенный охлажденный воздух.

Поверхность платтера имеет **магнитное покрытие** толщиной всего лишь в 1,1 мкм, а также **слой смазки** для предохранения головки от повреждения при опускании и подъеме на ходу. При вращении платтера над ним образуется **воздушный слой**, который обеспечивает воздушную подушку для зависания головки на высоте 0,5 мкм над поверхностью диска.

Винчестерские накопители имеют очень большую ёмкость: от 10 до 100 Гбайт. У современных моделей скорость вращения шпинделя (вращающего вала) обычно составляет 7200 об/мин, среднее время поиска данных 9 мс, средняя скорость передачи данных до 60 Мбайт/с. В отличие от дискеты, жесткий диск **вращается непрерывно**. Все современные накопители снабжаются **встроенным кэшем** (обычно 2 Мбайта), который существенно повышает их производительность. Винчестерский накопитель связан с процессором через **контроллер жесткого диска**.

3. Накопители на компакт-дисках



Здесь носителем информации является CD-ROM (Compact Disk Read-Only Memory - компакт диск, из которого можно только читать).

CD-ROM представляет собой прозрачный полимерный диск диаметром 12 см и толщиной 1,2 мм, на одну сторону которого напылен светоотражающий слой алюминия, защищенный от повреждений слоем прозрачного лака. Толщина напыления составляет несколько десятитысячных долей миллиметра.

Информация на диске представляется в виде последовательности **впадин** (углублений в диске) и **выступов** (их уровень соответствует поверхности диска), расположенных на спиральной дорожке, выходящей из области вблизи оси диска. На каждом дюйме (2,54 см) по радиусу диска размещается 16 тысяч витков спиральной дорожки. Для сравнения — на поверхности жесткого диска на дюйме по радиусу

помещается лишь несколько сотен дорожек. Емкость CD достигает **780 Мбайт**. Информация наносится на диск при его изготовлении и не может быть изменена.

CD-ROM обладают высокой удельной информационной емкостью, что позволяет создавать на их основе справочные системы и учебные комплексы с большой иллюстративной базой. Один CD по информационной емкости равен почти 500 дискетам. Считывание информации с CD-ROM происходит с достаточно высокой скоростью, хотя и заметно меньшей, чем скорость работы накопителей на жестком диске. CD-ROM просты и удобны в работе, имеют низкую удельную стоимость хранения данных, практически не изнашиваются, не могут быть поражены вирусами, с них невозможно случайно стереть информацию.

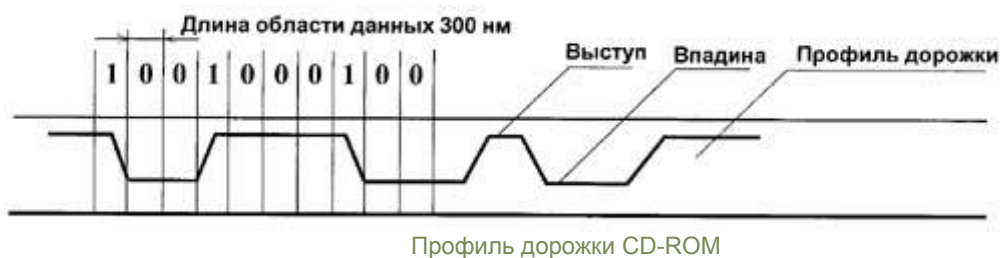
В отличие от магнитных дисков, компакт-диски имеют не множество кольцевых дорожек, а **одну — спиральную**, как у грампластинок. В связи с этим, угловая скорость вращения диска не постоянна. Она линейно уменьшается в процессе продвижения читающей лазерной головки к краю диска.



Рис. 2.9. Накопитель CD-ROM

Для работы с CD-ROM нужно подключить к компьютеру **накопитель CD-ROM** (рис. 2.9), преобразующий последовательность углублений и выступов на поверхности CD-ROM в последовательность двоичных сигналов. Для этого используется **считывающая головка с микролазером и светодиодом**. Глубина впадин на поверхности диска равна четверти длины волны лазерного света. Если в двух последовательных тактах считывания информации луч света лазерной головки переходит с выступа на дно впадины или обратно, разность длин путей света в этих тактах меняется на полуволну, что вызывает усиление или ослабление совместно попадающих на светодиод прямого и отраженного от диска света.

Если в последовательных тактах считывания длина пути света не меняется, то и состояние светодиода не меняется. В результате ток через светодиод образует последовательность двоичных электрических сигналов, соответствующих сочетанию впадин и выступов на дорожке.



> Различная длина оптического пути луча света в двух последовательных тактах считывания информации соответствует двоичным единицам. Одинаковая длина соответствует двоичным нулям.

Сегодня почти все персональные компьютеры имеют накопитель CD-ROM. Но многие мультимедийные интерактивные программы слишком велики, чтобы поместиться на одном CD. **На смену технологии CD-ROM стремительно идет технология цифровых видеодисков DVD.** Эти диски имеют тот же размер, что и обычные CD, но вмещают до **17 Гбайт данных**, т.е. по объему заменяют 20 стандартных дисков CD-ROM. На таких дисках выпускаются **мультимедийные игры и интерактивные видеофильмы** отличного качества, позволяющие зрителю просматривать эпизоды под разными углами камеры, выбирать различные варианты окончания картины, знакомиться с биографиями снявшихся актеров, наслаждаться великолепным качеством звука.

4. Записывающие оптические и магнитооптические накопители

- **Записывающий накопитель CD-R** (Compact Disk Recordable) способен, наряду с прочтением обычных компакт-дисков, записывать информацию на специальные оптические диски емкостью 650 Мбайт. В дисках CD-R отражающий слой выполнен из золотой пленки. Между этим слоем и поликарбонатной основой расположен регистрирующий слой из органического материала, темнеющего при нагревании. В процессе записи лазерный луч нагревает выбранные точки слоя, которые темнеют и перестают пропускать свет к отражающему слою, образуя участки, аналогичные впадинам. Накопители CD-R, благодаря сильному удешевлению, приобретают все большее распространение.



Рис.2.10. Накопитель CD-MO

- **Накопитель на магнито-оптических компакт-дисках CD-MO** (Compact Disk — Magneto Optical) (рис. 2.10). Диски CD-MO можно многократно использовать для записи. Ёмкость от 128 Мбайт до 2,6 Гбайт.

- **Записывающий накопитель CD-R** (Compact Disk Recordable) способен, наряду с прочтением обычных компакт-дисков, записывать информацию на специальные оптические диски. Ёмкость 650 Мбайт.

- **Накопитель WARM** (Write And Read Many times), позволяет производить многократную запись и считывание.

5. Накопители на магнитной ленте (стримеры) и накопители на сменных дисках

Стример (англ. tape streamer) — устройство для резервного копирования больших объёмов информации. В качестве носителя здесь применяются кассеты с магнитной лентой ёмкостью 1 — 2 Гбайта и больше.



Рис. 2.11. Накопитель на сменных дисках

Стримеры позволяют записать на небольшую кассету с магнитной лентой огромное количество информации. Встроенные в стример средства аппаратного сжатия позволяют автоматически уплотнять информацию перед её записью и восстанавливать после считывания, что увеличивает объём сохраняемой информации.

Недостатком стримеров является их сравнительно низкая скорость записи, поиска и считывания информации.

В последнее время всё шире используются **накопители на сменных дисках**, которые позволяют не только увеличивать объём хранимой информации, но и переносить информацию между компьютерами. Объём сменных дисков — от сотен Мбайт до нескольких Гигабайт.

2.11. Что такое аудиоадаптер?

Аудиоадаптер (Sound Blaster или звуковая плата) это специальная электронная плата, которая позволяет записывать звук, воспроизводить его и создавать программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

Аудиоадаптер содержит в себе два преобразователя информации:

- **аналого-цифровой**, который преобразует непрерывные (то есть, аналоговые) звуковые сигналы (речь, музыку, шум) в цифровой двоичный код и записывает его на магнитный носитель;
- **цифро-аналоговый**, выполняющий обратное преобразование сохранённого в цифровом виде звука в аналоговый сигнал, который затем воспроизводится с помощью акустической системы, синтезатора звука или наушников.

Профессиональные звуковые платы позволяют выполнять сложную обработку звука, обеспечивают стереозвучание, имеют собственное ПЗУ с хранящимися в нём сотнями тембров звучаний различных музыкальных инструментов. Звуковые файлы обычно имеют очень большие размеры. Так, трёхминутный звуковой файл со стереозвучанием занимает примерно 30 Мбайт памяти. Поэтому **платы Sound Blaster, помимо своих основных функций, обеспечивают автоматическое сжатие файлов.**

Область применения звуковых плат — компьютерные игры, обучающие программные системы, рекламные презентации, "голосовая почта" (voice mail) между компьютерами, озвучивание различных процессов, происходящих в компьютерном оборудовании, таких, например, как отсутствие бумаги в принтере и т.п.

2.12. Что такое видеоадаптер и графический акселератор?

Видеоадаптер — это электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея. Содержит **видеопамять, регистры ввода вывода и модуль BIOS**. Посылает в дисплей сигналы управления яркостью лучей и сигналы развертки изображения.

Наиболее распространенный видеоадаптер на сегодняшний день — **адаптер SVGA** (Super Video Graphics Array — супервидеографический массив), **который может отображать на экране дисплея 1280x1024 пикселей при 256 цветах и 1024x768 пикселей при 16 миллионах цветов.**

С увеличением числа приложений, использующих сложную графику и видео, наряду с традиционными видеоадаптерами широко используются разнообразные **устройства компьютерной обработки видеосигналов:**



Рис. 2.12. Графический акселератор

- **Графические акселераторы** (ускорители) — специализированные графические сопроцессоры, увеличивающие эффективность видеосистемы. Их применение освобождает центральный процессор от большого объема операций с видеоданными, так как акселераторы самостоятельно вычисляют, какие пиксели отображать на экране и каковы их цвета.

- **Фрейм-грабберы**, которые позволяют отображать на экране компьютера видеосигнал от видеомagneтофона, камеры, лазерного проигрывателя и т. п., с тем, чтобы захватить нужный кадр в память и впоследствии сохранить его в виде файла.

- **TV-тюнеры** — видеоплаты, превращающие компьютер в телевизор. TV-тюнер позволяет выбрать любую нужную телевизионную программу и отображать ее на экране в масштабируемом окне. Таким образом можно следить за ходом передачи, не прекращая работу.

2.13. Что такое клавиатура?

Клавиатура компьютера — устройство для ввода информации в компьютер и подачи управляющих сигналов. Содержит стандартный набор клавиш печатной машинки и некоторые дополнительные клавиши — управляющие и функциональные клавиши, клавиши управления курсором и малую цифровую клавиатуру.

Все символы, набираемые на клавиатуре, немедленно отображаются на мониторе в позиции курсора (курсoр — светящийся символ на экране монитора, указывающий позицию, на которой будет отображаться следующий вводимый с клавиатуры знак).

Наиболее распространена сегодня клавиатура с раскладкой клавиш **QWERTY** (читается "кверти"), названная так по клавишам, расположенным в верхнем левом ряду алфавитно-цифровой части клавиатуры:



Рис. 2.13. Клавиатура компьютера

Такая клавиатура имеет **12 функциональных клавиш**, расположенных вдоль верхнего края. Нажатие функциональной клавиши приводит к посылке в компьютер не одного символа, а целой совокупности символов. Функциональные клавиши могут программироваться пользователем. Например, во многих программах для получения помощи (подсказки) задействована клавиша **F1**, а для выхода из программы — клавиша **F10**.

Управляющие клавиши имеют следующее назначение:

- **Enter** — клавиша **ввода**;
- **Esc** (Escape — выход) клавиша для **отмены** каких-либо действий, выхода из программы, из меню и т.п.;
- **Ctrl** и **Alt** — эти клавиши самостоятельного значения не имеют, но при нажатии совместно с другими управляющими клавишами изменяют их действие;
- **Shift** (регистр) — обеспечивает **смену регистра клавиш** (верхнего на нижний и наоборот);
- **Insert** (вставлять) — **переключает режимы вставки** (новые символы вводятся посреди уже набранных, раздвигая их) и **замены** (старые символы замещаются новыми);
- **Delete** (удалять) — **удаляет символ** с позиции курсора;
- **Back Space** или ← — **удаляет символ** перед курсором;
- **Home** и **End** — обеспечивают **перемещение курсора в первую и последнюю позицию строки**, соответственно;
- **Page Up** и **Page Down** — обеспечивают **перемещение по тексту на одну страницу** (один экран) назад и вперед, соответственно;

- **Tab** — клавиша табуляции, обеспечивает перемещение курсора вправо сразу на несколько позиций до очередной позиции табуляции;
- **Caps Lock** — фиксирует верхний регистр, обеспечивает **ввод прописных букв вместо строчных**;
- **Print Screen** — обеспечивает **печать информации**, видимой в текущий момент на экране.
- **Длинная нижняя клавиша** без названия — предназначена для **ввода пробелов**.
- Клавиши **↑**, **↓**, **←** и **→** служат для перемещения курсора **вверх, вниз, влево и вправо** на одну позицию или строку.

Малая цифровая клавиатура используется в двух режимах — **ввода чисел и управления курсором**. Переключение этих режимов осуществляется клавишей **Num Lock**.

Клавиатура содержит встроенный **микроконтроллер** (местное устройство управления), который выполняет следующие функции:

- последовательно опрашивает клавиши, считывая введенный сигнал и вырабатывая двоичный **скан-код** клавиши;
- управляет световыми индикаторами клавиатуры;
- проводит внутреннюю диагностику неисправностей;
- осуществляет взаимодействие с центральным процессором через **порт ввода-вывода** клавиатуры.

Клавиатура имеет **встроенный буфер** — промежуточную память малого размера, куда помещаются введенные символы. В случае переполнения буфера нажатие клавиши будет сопровождаться звуковым сигналом — это означает, что символ не введен (отвергнут). Работу клавиатуры поддерживают специальные программы, "защитые" в **BIOS**, а также **драйвер** клавиатуры, который обеспечивает возможность ввода русских букв, управление скоростью работы клавиатуры и др.

2.14. Что такое видеосистема компьютера?

Видеосистема компьютера состоит из трех компонент:

- **монитор** (называемый также дисплеем);
- **видеоадаптер**;
- **программное обеспечение** (драйверы видеосистемы).

Видеоадаптер посылает в монитор сигналы управления яркостью лучей и синхросигналы строчной и кадровой разверток. **Монитор** преобразует эти сигналы в зрительные образы. А **программные средства** обрабатывают видеоизображения — выполняют кодирование и декодирование сигналов, координатные преобразования, сжатие изображений и др.

Монитор — устройство визуального отображения информации (в виде текста, таблиц, рисунков, чертежей и др.).



Рис. 2.14. Монитор

Подавляющее большинство мониторов сконструированы на базе **электронно-лучевой трубки (ЭЛТ)**, и принцип их работы аналогичен принципу работы телевизора. Мониторы бывают алфавитно-цифровые и графические, монохромные и цветного изображения. Современные компьютеры комплектуются, как правило, цветными графическими мониторами.

1. Монитор на базе электронно-лучевой трубки

Основной элемент дисплея — **электронно-лучевая трубка**. Её передняя, обращенная к зрителю часть с внутренней стороны покрыта **люминофором** — специальным веществом, способным излучать свет при попадании на него быстрых электронов.

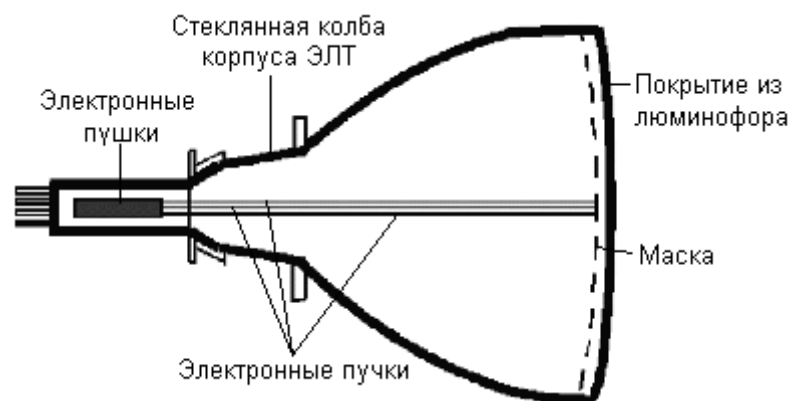


Рис. 2.15. Схема электронно-лучевой трубки

Люминофор наносится в виде наборов точек трёх основных цветов — **красного**, **зелёного** и **синего**. Эти цвета называют основными, потому что их сочетаниями (в различных пропорциях) можно представить любой цвет спектра.

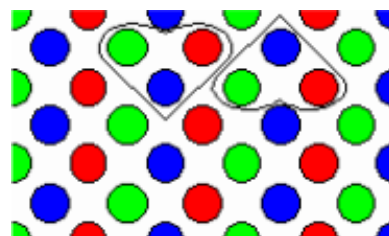


Рис. 2.16. Пиксельные триады

Наборы точек люминофора располагаются по треугольным триадам. Триада образует **пиксел** — точку, из которых формируется изображение (англ. pixel — picture element,

элемент картинки).

Расстояние между центрами пикселей называется **точечным шагом монитора**. Это расстояние существенно влияет на чёткость изображения. Чем меньше шаг, тем выше чёткость. Обычно в цветных мониторах шаг составляет 0,24 мм. При таком шаге глаз человека воспринимает точки триады как одну точку "сложного" цвета.

На противоположной стороне трубки расположены три (по количеству основных цветов) **электронные пушки**. Все три пушки "нацелены" на один и тот же пиксел, но каждая из них излучает поток электронов в сторону "своей" точки люминофора. Чтобы электроны беспрепятственно достигали экрана, из трубки откачивается воздух, а между пушками и экраном создаётся высокое электрическое напряжение, ускоряющее электроны. Перед экраном на пути электронов ставится **маска** — тонкая металлическая пластина с большим количеством отверстий, расположенных напротив точек люминофора. Маска обеспечивает попадание электронных лучей только в точки люминофора соответствующего цвета.

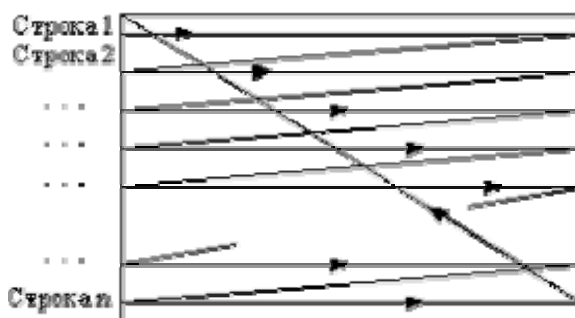


Рис. 2.17. Ход электронного пучка по экрану

Величиной электронного тока пушек и, следовательно, яркостью свечения пикселей, управляет сигнал, поступающий с видеоадаптера.

На ту часть колбы, где расположены электронные пушки, надевается **отклоняющая система** монитора, которая заставляет электронный пучок пробегать поочерёдно все пиксели строчку за строчкой от верхней до нижней, затем возвращаться в начало верхней строки и т.д.

Количество отображённых строк в секунду называется **строчной частотой развертки**. А частота, с которой меняются кадры изображения, называется **кадровой частотой развёртки**. Последняя не должна быть ниже 85 Гц, иначе изображение будет мерцать.

2. Жидкокристаллические мониторы

Все шире используются наряду с традиционными ЭЛТ-мониторами. **Жидкие кристаллы** — это особое состояние некоторых органических веществ, в котором они обладают текучестью и свойством образовывать пространственные структуры, подобные кристаллическим. Жидкие кристаллы могут изменять свою структуру и светооптические свойства под действием электрического напряжения. Меняя с помощью электрического

поля ориентацию групп кристаллов и используя введенные в жидкокристаллический раствор вещества, способные излучать свет под воздействием электрического поля, можно создать высококачественные изображения, передающие более 15 миллионов цветовых оттенков.

Большинство ЖК-мониторов использует тонкую плёнку из жидких кристаллов, помещённую между двумя стеклянными пластинами. Заряды передаются через так называемую **пассивную матрицу** — сетку невидимых нитей, горизонтальных и вертикальных, создавая в месте пересечения нитей точку изображения (несколько размытого из-за того, что заряды проникают в соседние области жидкости).



Рис. 2.18. Жидкокристаллический монитор

Активные матрицы вместо нитей используют прозрачный экран из транзисторов и обеспечивают яркое, практически не имеющее искажений изображение. Экран при этом разделен на независимые ячейки, каждая из которых состоит из четырех частей (для трёх основных цветов и одна резервная). Количество таких ячеек по широте и высоте экрана называют **разрешением экрана**. Современные ЖК-мониторы имеют разрешение 642x480, 1280x1024 или 1024x768. Таким образом, экран имеет от 1 до 5 млн точек, каждая из которых управляется собственным транзистором. По компактности такие мониторы не знают себе равных. Они занимают в 2 — 3 раза меньше места, чем мониторы с ЭЛТ и во столько же раз легче; потребляют гораздо меньше электроэнергии и не излучают электромагнитных волн, воздействующих на здоровье людей.

3. Сенсорный экран



Рис. 2.19. Сенсорный экран

Общение с компьютером осуществляется путём прикосновения пальцем к определённому месту чувствительного экрана. Этим выбирается необходимый режим из меню, показанного на экране монитора. (**Меню** — это выведенный на экран монитора список различных вариантов работы компьютера, по которому можно сделать конкретный выбор.) Сенсорными экранами оборудуют рабочие места операторов и диспетчеров, их используют в **информационно-справочных системах** и т.д.

2.15. Что такое принтер, плоттер, сканер?

Принтер — печатающее устройство. Осуществляет вывод из компьютера закодированной информации в виде печатных копий текста или графики.

Существуют тысячи наименований принтеров. Но основных видов принтеров три: матричные, лазерные и струйные.



- **Матричные принтеры** используют комбинации маленьких штырьков, которые бьют по красящей ленте, благодаря чему на бумаге остаётся отпечаток символа. Каждый символ, печатаемый на принтере, формируется из набора 9, 18 или 24 игл, сформированных в виде вертикальной колонки. Недостатками этих недорогих принтеров являются их шумная работа и невысокое качество печати.

- **Лазерные принтеры** работают примерно так же, как ксероксы. Компьютер формирует в своей памяти "образ" страницы текста и передает его принтеру. Информация о странице проецируется с помощью лазерного луча на вращающийся барабан со светочувствительным покрытием, меняющим электрические свойства в зависимости от освещённости.



Лазерный принтер

После засветки на барабан, находящийся под электрическим напряжением, наносится красящий порошок — **тонер**, частицы которого налипают на засвеченные участки поверхности барабана. Принтер с помощью специального горячего валика протягивает бумагу под барабаном; тонер переносится на бумагу и "вплавляется" в неё, оставляя стойкое высококачественное изображение. **Цветные** лазерные принтеры пока очень дороги.

- **Струйные принтеры** генерируют символы в виде последовательности **чернильных точек**. Печатающая головка принтера имеет крошечные **сопла**, через которые на страницу выбрызгиваются быстросохнущие чернила. Эти принтеры требовательны к качеству бумаги. **Цветные** струйные принтеры создают цвета, комбинируя чернила **четырёх** основных цветов — **ярко-голубого, пурпурного, желтого и черного**.

Принтер связан с компьютером посредством **кабеля** принтера, один конец которого вставляется своим разъёмом в **гнездо** принтера, а другой — в **порт** принтера компьютера. **Порт** — это **разъём, через который можно соединить процессор компьютера с внешним устройством**.

Каждый принтер обязательно имеет свой **драйвер** — программу, которая способна переводить (транслировать) стандартные команды печати компьютера в специальные команды, требующиеся для каждого принтера.

Плоттер (графопостроитель) — устройство, которое чертит графики, рисунки или диаграммы под управлением компьютера.



Роликовый плоттер

Плоттеры используются для получения сложных конструкторских чертежей, архитектурных планов, географических и метеорологических карт, деловых схем. **Плоттеры рисуют изображения с помощью пера.**

Роликовые плоттеры прокручивают бумагу под пером, а **планшетные плоттеры** перемещают перо через всю поверхность горизонтально лежащей бумаги.

Плоттеру, так же, как и принтеру, обязательно нужна специальная программа — **драйвер**, позволяющая прикладным программам передавать ему инструкции: поднять и опустить перо, провести линию заданной толщины и т.п.

Сканер — устройство для ввода в компьютер графических изображений. Создает оцифрованное изображение документа и помещает его в память компьютера.



Планшетный сканер

Если принтеры выводят информацию из компьютера, то сканеры, наоборот, **переносят информацию с бумажных документов в память компьютера**. Существуют **ручные сканеры**, которые прокатывают по поверхности документа рукой, и **планшетные сканеры**, по внешнему виду напоминающие копировальные машины.

Если при помощи сканера вводится текст, компьютер воспринимает его как картинку, а не как последовательность символов. Для преобразования такого графического текста в обычный символьный формат используют программы оптического распознавания образов.

2.16. Что такое модем и факс-модем?

Модем — устройство для передачи компьютерных данных на большие расстояния по телефонным линиям связи.

Цифровые сигналы, вырабатываемые компьютером, нельзя напрямую передавать по телефонной сети, потому что она предназначена для передачи человеческой речи — непрерывных сигналов звуковой частоты.

Модем обеспечивает преобразование цифровых сигналов компьютера в переменный ток частоты звукового диапазона — этот процесс называется **модуляцией**, а также обратное преобразование, которое называется **демодуляцией**. Отсюда название устройства: модем — **модулятор/демодулятор**.



Рис. 2.24. Схема реализации модемной связи

Для осуществления связи один модем вызывает другой по номеру телефона, а тот отвечает на вызов. Затем модемы посылают друг другу сигналы, согласуя подходящий им обоим **режим связи**. После этого передающий модем начинает **посылать модулированные данные** с согласованной скоростью (количеством бит в секунду) и форматом. Модем на другом конце **преобразует полученную информацию в цифровой вид** и передает её своему компьютеру. Закончив сеанс связи, модем отключается от линии.



Рис. 2.25 Внешний модем

Управление модемом осуществляется с помощью специального **коммутационного программного обеспечения**.

Модемы бывают **внешние**, выполненные в виде отдельного устройства, и **внутренние**, представляющие собой электронную плату, устанавливаемую внутри компьютера. Почти все модемы поддерживают и функции факсов.

Факс — это устройство факсимильной передачи изображения по телефонной сети. Название "факс" произошло от слова "факсимиле" (лат. fac simile — сделай подобное), означающее точное воспроизведение графического оригинала (подписи, документа и т.д.) средствами печати. Модем, который может передавать и получать данные как факс, называется **факс-модемом**.

2.17. Что такое манипуляторы?

Манипуляторы (мышь, джойстик и др.) — это специальные устройства, которые используются для **управления курсором**.



Мышь имеет вид небольшой коробки, полностью уместящейся на ладони. Мышь связана с компьютером кабелем через специальный блок — адаптер, и её движения преобразуются в соответствующие перемещения курсора по экрану дисплея. В верхней части устройства расположены управляющие кнопки (обычно их три), позволяющие задавать начало и конец движения, осуществлять выбор меню и т.п.



Джойстик — обычно это стержень-ручка, отклонение которой от вертикального положения приводит к передвижению курсора в соответствующем направлении по экрану монитора. Часто применяется в компьютерных играх. В некоторых моделях в джойстик монтируется датчик давления. В этом случае, чем сильнее пользователь нажимает на ручку, тем быстрее движется курсор по экрану дисплея.



Трекбол — небольшая коробочка с шариком, встроенным в верхнюю часть корпуса. Пользователь рукой вращает шарик и перемещает, соответственно, курсор. В отличие от мыши, трекбол не требует свободного пространства около компьютера, его можно встроить в корпус машины.

Дигитайзер — устройство для преобразования готовых изображений (чертежей, карт) в цифровую форму. Представляет собой плоскую панель — **планшет**, располагаемую на столе, и специальный инструмент — **перо**, с помощью которого указывается позиция на планшете. При перемещении пера по планшету фиксируются его координаты в близко расположенных точках, которые затем преобразуются в компьютере в требуемые единицы измерения.

2.18. Как устроен компьютер?

Рассмотрим устройство компьютера на примере самой распространенной компьютерной системы — персонального компьютера. **Персональным компьютером** (ПК) называют сравнительно недорогой универсальный микрокомпьютер, рассчитанный на одного пользователя. Персональные компьютеры обычно проектируются на основе принципа открытой архитектуры.

Принцип открытой архитектуры заключается в следующем:

- Регламентируются и стандартизируются только описание принципа действия компьютера и его конфигурация (определенная совокупность аппаратных средств и соединений между ними). Таким образом, компьютер можно собирать из отдельных узлов и деталей, разработанных и изготовленных независимыми фирмами-изготовителями.
- Компьютер легко расширяется и модернизируется за счёт наличия внутренних расширительных гнезд, в которые пользователь может вставлять разнообразные устройства, удовлетворяющие заданному стандарту, и тем самым устанавливать конфигурацию своей машины в соответствии со своими личными предпочтениями.

Упрощённая блок-схема, отражающая основные функциональные компоненты компьютерной системы в их взаимосвязи, изображена на рисунке 2.26.

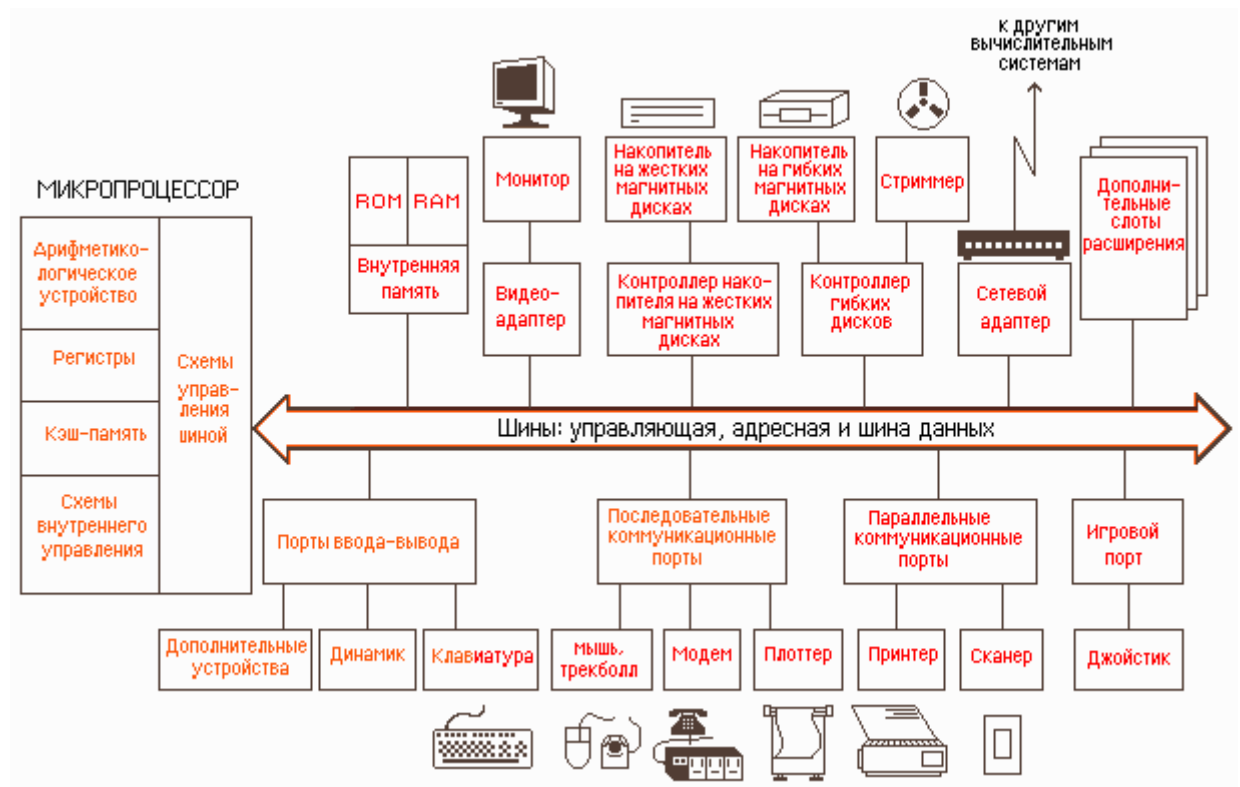


Рис. 2.26. Общая структура персонального компьютера с подсоединенными периферийными устройствами

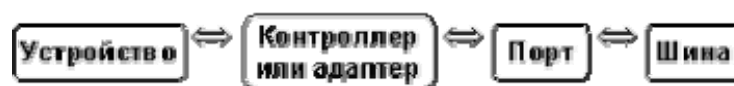
Для того, чтобы соединить друг с другом различные устройства компьютера, они должны иметь одинаковый **интерфейс** (англ. interface от inter — между, и face — лицо).

Интерфейс — это средство сопряжения двух устройств, в котором все физические и логические параметры согласуются между собой.

Если интерфейс является общепринятым, например, утверждённым на уровне международных соглашений, то он называется **стандартным**.

Каждый из функциональных элементов (память, монитор или другое устройство) связан с шиной определённого типа — адресной, управляющей или шиной данных.

Для согласования интерфейсов периферийные устройства подключаются к шине не напрямую, а через свои **контроллеры** (адаптеры) и **порты** примерно по такой схеме:



Контроллеры и адаптеры представляют собой наборы электронных цепей, которыми снабжаются устройства компьютера с целью совместимости их интерфейсов. Контроллеры, кроме этого, осуществляют непосредственное управление периферийными устройствами по запросам микропроцессора.

Порты устройств представляют собой некие электронные схемы, содержащие один или несколько регистров ввода-вывода и позволяющие подключать периферийные устройства

компьютера к внешним шинам микропроцессора.

Портами также называют **устройства стандартного интерфейса**: последовательный, параллельный и игровой порты (или интерфейсы).

Последовательный порт обменивается данными с процессором побайтно, а с внешними устройствами — побитно. Параллельный порт получает и посылает данные побайтно.

К **последовательному** порту обычно подсоединяют медленно действующие или достаточно удалённые устройства, такие, как мышь и модем. К **параллельному** порту подсоединяют более "быстрые" устройства — принтер и сканер. Через **игровой** порт подсоединяется джойстик. Клавиатура и монитор подключаются к своим **специализированным** портам, которые представляют собой просто **разъёмы**.

Основные электронные компоненты, определяющие архитектуру процессора, размещаются на основной плате компьютера, которая называется **системной** или **материнской** (**MotherBoard**). А контроллеры и адаптеры дополнительных устройств, либо сами эти устройства, выполняются в виде **плат расширения** (DaughterBoard — дочерняя плата) и подключаются к шине с помощью **разъёмов расширения**, называемых также **слотами расширения** (англ. slot — щель, паз).

2.19. Какие основные блоки входят в состав компьютера?

Современный персональный компьютер состоит из нескольких основных конструктивных компонент:

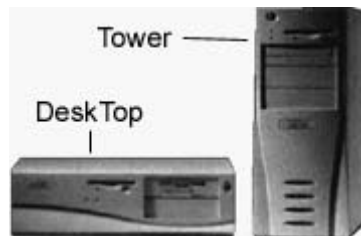


Рис. 2.27. Виды корпусов системного блока

- системного блока;
- монитора;
- клавиатуры;
- манипуляторов.

В системном блоке размещаются:

- блок питания;
- накопитель на жёстких магнитных дисках;
- накопитель на гибких магнитных дисках;
- системная плата;
- платы расширения;
- накопитель CD-ROM;

- и др.

Корпус системного блока может иметь горизонтальную (DeskTop) или вертикальную (Tower — башня) компоновку. Типичный системный блок со снятой крышкой корпуса — на рис. 2.28.

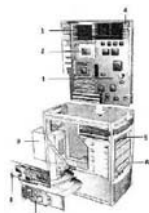


Рис. 2.28. Системный блок со снятой крышкой корпуса

- 1 — Системная плата.
- 2 — Разъём дополнительного второго процессора.
- 3 — Центральный процессор с радиатором для отвода тепла.
- 4 — Разъёмы оперативной памяти.
- 5 — Накопитель на гибких магнитных дисках.
- 6 — Накопитель CD-ROM.
- 7 — Сетевая карта.
- 8 — Графический акселератор.
- 9 — Блок питания, преобразующий переменное напряжение электросети в постоянное напряжение различной полярности и величины, необходимое для питания системной платы и внутренних устройств. Блок питания содержит вентилятор, создающий циркулирующие потоки воздуха для охлаждения системного блока.

2.20. Что собой представляет системная плата?

Системная плата является основной в системном блоке. Она содержит компоненты, определяющие архитектуру компьютера:

- центральный процессор;
- постоянную ([ROM](#)) и оперативную ([RAM](#)) память, [кэш-память](#);
- интерфейсные схемы шин;
- гнезда расширения;
- обязательные системные средства ввода-вывода и др.

Системные платы исполняются на основе наборов микросхем, которые называются **чипсетами** (ChipSets). Часто на системных платах устанавливают и контроллеры дисковых накопителей, видеоадаптер, контроллеры портов и др. В гнезда расширения системной платы устанавливаются платы таких периферийных устройств, как модем, сетевая плата, видеоплата и т.п.

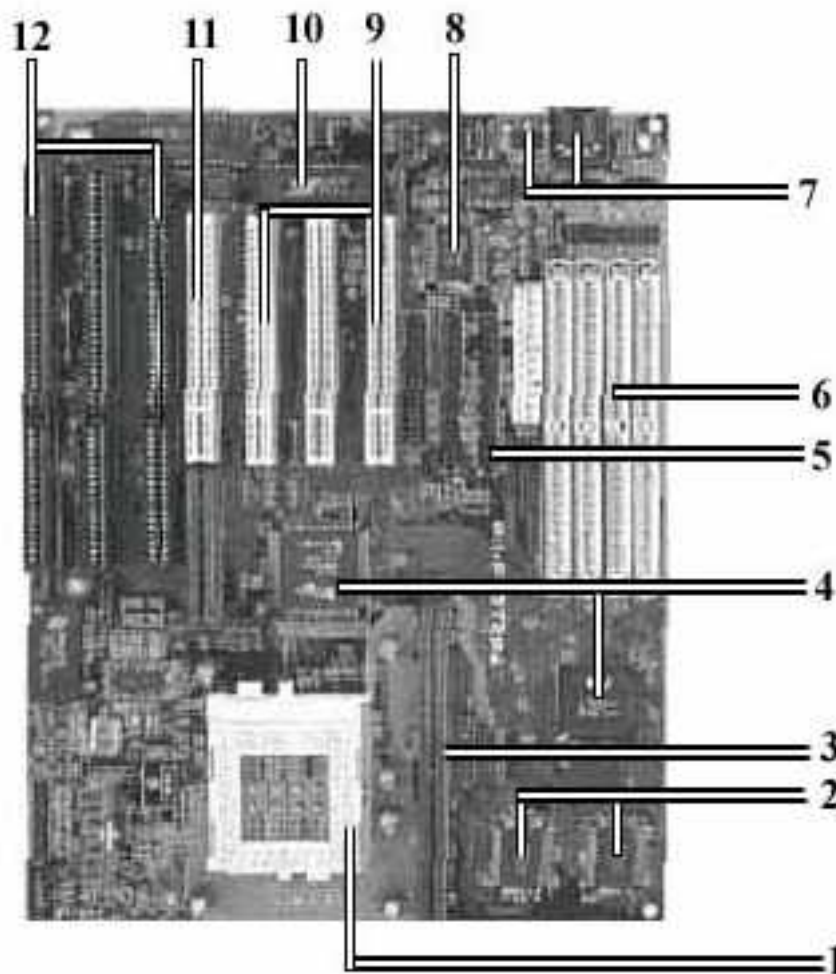


Рис. 2.29. Системная плата компьютера класса Pentium

- 1 — Разъём под центральный процессор;
- 2 — Дополнительный кэш объёмом 256 Кбайт;
- 3 — Разъём под дополнительный кэш;
- 4 — Контроллеры внешних устройств;
- 5 — Разъёмы накопителей на жёстких магнитных дисках;
- 6 — Разъёмы под оперативную память, 4 планки;
- 7 — Коннектор (соединитель) клавиатуры и мыши;
- 8 — Микросхема, обслуживающая флоппи-дисковод, последовательные порты и параллельный порт;
- 9 — Разъёмы 32-битной шины (для видеокарты, карты Интернет и др.);
- 10 — Перезаписываемая BIOS (Flash-память);
- 11 — Мультимедийная шина;
- 12 — Разъёмы 16-битной шины.

2.21. Как организуется межкомпьютерная связь?

Назовём задачи, которые трудно или невозможно решить без организации информационной связи между различными компьютерами:

- **перенос информации на большие расстояния** (сотни, тысячи километров);
- **совместное использование несколькими компьютерами дорогостоящих аппаратных, программных или информационных ресурсов** — мощного процессора, ёмкого накопителя, высокопроизводительного лазерного принтера, баз данных, программного обеспечения и т.д.;
- перенос информации с одного компьютера на другой **при несовместимых флоппи-дисководах** (5,25 и 3,5 дюйма);
- **совместная работа над большим проектом**, когда исполнители должны всегда иметь последние (актуальные) копии общих данных во избежание путаницы, и т.д.

Есть три основных способа организации межкомпьютерной связи:

- объединение двух рядом расположенных компьютеров через их коммуникационные порты посредством специального **кабеля**;

- передача данных от одного компьютера к другому посредством **модема** с помощью проводных или спутниковых линий связи;
- объединение компьютеров в **компьютерную сеть**.



Часто при организации связи между двумя компьютерами за **одним компьютером закрепляется роль поставщика ресурсов** (программ, данных и т.д.), а за другим — **роль пользователя этих ресурсов**. В этом случае первый компьютер называется **сервером**, а второй — **клиентом** или рабочей станцией. Работать можно только на компьютере-клиенте под управлением специального программного обеспечения.

Сервер (англ. serve — обслуживать) — это высокопроизводительный компьютер с большим объёмом внешней памяти, который обеспечивает обслуживание других компьютеров путем управления распределением дорогостоящих ресурсов совместного пользования (программ, данных и периферийного оборудования).

Клиент (иначе, рабочая станция) — любой компьютер, имеющий доступ к услугам сервера.



Сетевой сервер
HP LD PRO

Например, сервером может быть мощный компьютер, на котором размещается центральная база данных, а клиентом — обычный компьютер, программы которого по мере необходимости запрашивают данные с сервера. В некоторых случаях компьютер может быть **одновременно и клиентом, и сервером**. Это значит, что он может предоставлять свои ресурсы и хранимые данные другим компьютерам и одновременно использовать их ресурсы и данные.

Клиентом также называют **прикладную программу**, которая от имени пользователя **получает услуги сервера**. Соответственно, программное обеспечение, которое позволяет компьютеру **предоставлять услуги** другому компьютеру, называют **сервером** — так же, как и сам компьютер. Для преодоления **несовместимости интерфейсов** отдельных компьютеров вырабатывают специальные стандарты, называемые протоколами коммуникации.

Протокол коммуникации — это согласованный набор конкретных правил обмена информацией между разными устройствами передачи данных. Имеются протоколы для скорости передачи, форматов данных, контроля ошибок и др.

Для работы с сетью необходимо наличие специального **сетевого программного обеспечения**, которое обеспечивает передачу данных в соответствии с заданным протоколом. Протоколы коммуникации предписывают разбить весь объём передаваемых данных на **пакеты** — отдельные блоки фиксированного размера. **Пакеты нумеруются**, чтобы их затем можно было собрать в правильной последовательности. К данным, содержащимся в пакете, добавляется дополнительная информация примерно такого формата:

Адрес получателя	Адрес отправителя	Длина	Данные	Поле контрольной суммы
------------------	-------------------	-------	--------	------------------------

Контрольная сумма данных пакета содержит информацию, необходимую для контроля ошибок. Первый раз она вычисляется передающим компьютером. После того, как пакет будет передан, контрольная сумма повторно вычисляется принимающим компьютером. Если значения не совпадают, это означает, что данные пакета были **повреждены при передаче**. Такой пакет отбрасывается, и автоматически направляется запрос **повторно передать пакет**.

При установлении связи устройства обмениваются сигналами для согласования коммуникационных каналов и протоколов. Этот процесс называется **подтверждением установления связи** (англ. HandShake — рукопожатие).

2.22. Что такое компьютерная сеть?

Компьютерная сеть (англ. Computer NetWork, от net — сеть и work — работа) — совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети.

Компьютерную сеть представляют как совокупность **узлов** (компьютеров и сетевого оборудования) и соединяющих их **ветвей** (каналов связи). **Ветвь сети** — это путь, соединяющий два смежных узла. Различают узлы **оконечные**, расположенные в конце только одной ветви, **промежуточные**, расположенные на концах более чем одной ветви, и **смежные** — такие узлы соединены по крайней мере одним путём, не содержащим никаких других узлов. Компьютеры могут объединяться в сеть разными способами.

Логический и физический способы соединения компьютеров, кабелей и других компонентов, в целом составляющих сеть, называется ее топологией. Топология характеризует свойства сетей, не зависящие от их размеров. При этом не учитывается производительность и принцип работы этих объектов, их типы, длины каналов, хотя при проектировании эти факторы очень важны.

СПРАВКА. Топология как математическое понятие:

Топология (от греч. *topos* — место и ... логия), раздел математики, изучающий топологические свойства фигур, т. е. свойства, не изменяющиеся при любых деформациях, производимых без разрывов и склеиваний. Примерами топологических свойств фигур являются размерность, число кривых, ограничивающих данную область и т. д. Так, окружность, эллипс, контур квадрата имеют одни и те же топологические свойства, т. к. эти линии могут быть деформированы одна в другую описанным выше образом; в то же время кольцо и круг обладают различными топологическими свойствами: круг ограничен одним контуром, а кольцо — двумя.
(Советский энциклопедический словарь. "Советская энциклопедия", 1979).

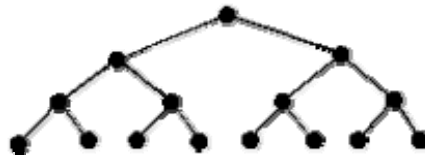
Наиболее распространенные виды топологий сетей:



Линейная сеть. Содержит только два конечных узла, любое число промежуточных узлов и имеет только один путь между любыми двумя узлами.



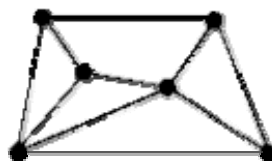
Кольцевая сеть. Сеть, в которой к каждому узлу присоединены две и только две ветви.



Древовидная сеть. Сеть, которая содержит более двух конечных узлов и по крайней мере два промежуточных узла, и в которой между двумя узлами имеется только один путь.



Звездообразная сеть. Сеть, в которой имеется только один промежуточный узел.



Ячеистая сеть. Сеть, которая содержит по крайней мере два узла, имеющих два или более пути между ними.

Полносвязанная сеть. Сеть, в которой имеется ветвь между любыми двумя узлами. Важнейшая характеристика компьютерной сети — её архитектура.

Архитектура сети — это реализованная структура сети передачи данных, определяющая её топологию, состав устройств и правила их взаимодействия в сети. В рамках архитектуры сети рассматриваются вопросы кодирования информации, её адресации и передачи, управления потоком сообщений, контроля ошибок и анализа работы сети в аварийных ситуациях и при ухудшении характеристик.

Наиболее распространённые архитектуры:

- **Ethernet** (англ. ether — эфир) — широковещательная сеть. Это значит, что все станции сети могут принимать все сообщения. Топология — линейная или звездообразная. Скорость передачи данных 10 или 100 Мбит/сек.
- **Arcnet** (Attached Resource Computer Network — компьютерная сеть соединённых ресурсов) — широковещательная сеть. Физическая топология — дерево. Скорость передачи данных 2,5 Мбит/сек.
- **Token Ring** (эстафетная кольцевая сеть, сеть с передачей маркера) — кольцевая сеть, в которой принцип передачи данных основан на том, что каждый узел кольца ожидает прибытия некоторой короткой уникальной последовательности битов — маркера — из смежного предыдущего узла. Поступление маркера указывает на то, что можно передавать сообщение из данного узла дальше по ходу потока. Скорость передачи данных 4 или 16 Мбит/сек.
- **FDDI** (Fiber Distributed Data Interface) — сетевая архитектура высокоскоростной передачи данных по оптоволоконным линиям. Скорость передачи — 100 Мбит/сек. Топология — двойное кольцо или смешанная (с включением звездообразных или древовидных подсетей). Максимальное количество станций в сети — 1000. Очень высокая стоимость оборудования.
- **ATM** (Asynchronous Transfer Mode) — перспективная, пока ещё очень дорогая архитектура, обеспечивает передачу цифровых данных, видеоинформации и голоса по одним и тем же линиям. Скорость передачи до 2,5 Гбит/сек. Линии связи оптические.

2.23. Как соединяются между собой устройства сети?

Для этого используется специальное оборудование:



Рис. 2.30. Сетевой интерфейсный адаптер

- **Сетевые кабели** (**коаксиальные**, состоящие из двух изолированных между собой концентрических проводников, из которых внешний имеет вид трубки; **оптоволоконные**; кабели на **витых парах**, образованные двумя переплетёнными друг с другом проводами, и др.).
- **Коннекторы** (соединители) для подключения кабелей к компьютеру; **разъёмы** для соединения отрезков кабеля.
- **Сетевые интерфейсные адаптеры** для приёма и передачи данных. В соответствии с определённым протоколом управляют доступом к среде передачи данных. Размещаются в системных блоках компьютеров, подключенных к сети. К разъёмам адаптеров подключается сетевой кабель.

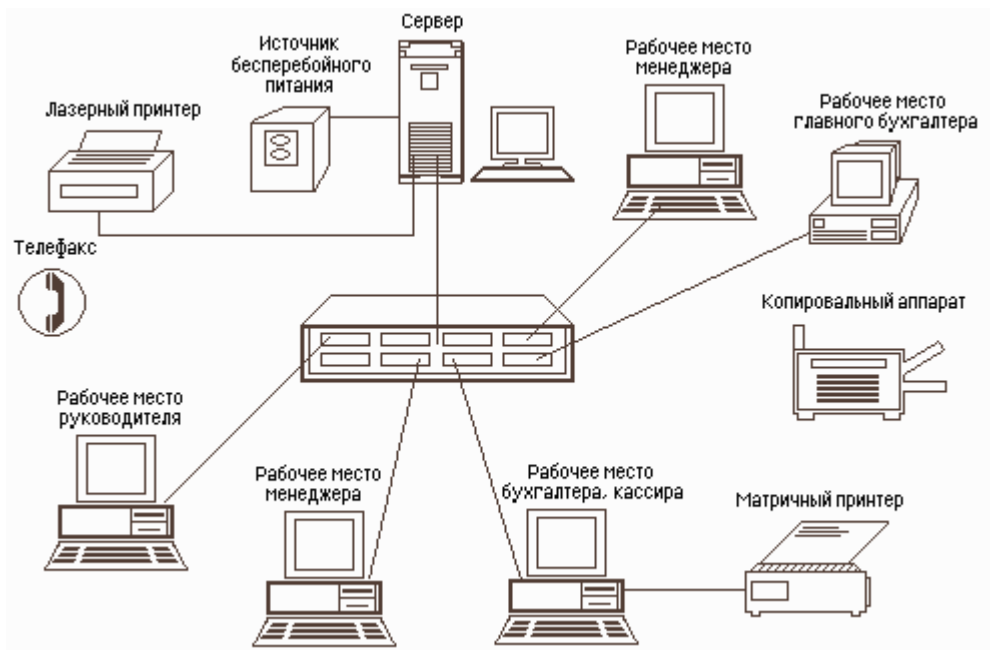


- **Трансиверы** повышают уровень качества передачи данных по кабелю, отвечают за приём сигналов из сети и обнаружение конфликтов.
- **Хабы** (концентраторы) и **коммутирующие хабы** (коммутаторы) расширяют топологические, функциональные и скоростные возможности компьютерных сетей. Хаб с набором разнотипных портов позволяет **объединять сегменты сетей с различными кабельными системами**. К порту хаба можно подключать как отдельный узел сети, так и другой хаб или сегмент кабеля.
- **Повторители** (репитеры) усиливают сигналы, передаваемые по кабелю при его большой длине.

2.24. Как классифицируют компьютерные сети по степени географического распространения?

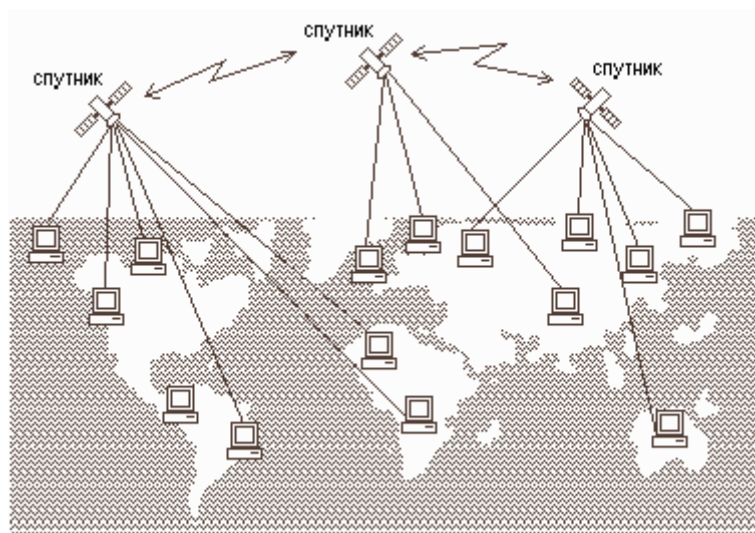
По степени географического распространения сети делятся на локальные, городские, корпоративные, глобальные и др.

Локальная сеть (ЛВС или LAN — Local Area NetWork) — сеть, связывающая ряд компьютеров в зоне, ограниченной пределами одной комнаты, здания или предприятия.



Небольшая офисная локальная сеть

Глобальная сеть (ГВС или WAN — World Area NetWork) — сеть, соединяющая компьютеры, удалённые географически на большие расстояния друг от друга. Отличается от локальной сети более протяженными коммуникациями (спутниковыми, кабельными и др.). Глобальная сеть объединяет локальные сети.



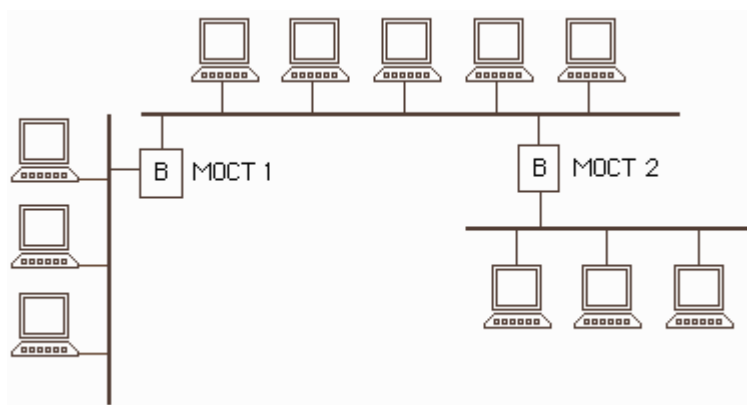
Глобальная сеть

Городская сеть (MAN — Metropolitan Area NetWork) — сеть, которая обслуживает информационные потребности большого города.

2.25. Как соединяются между собой локальные сети?

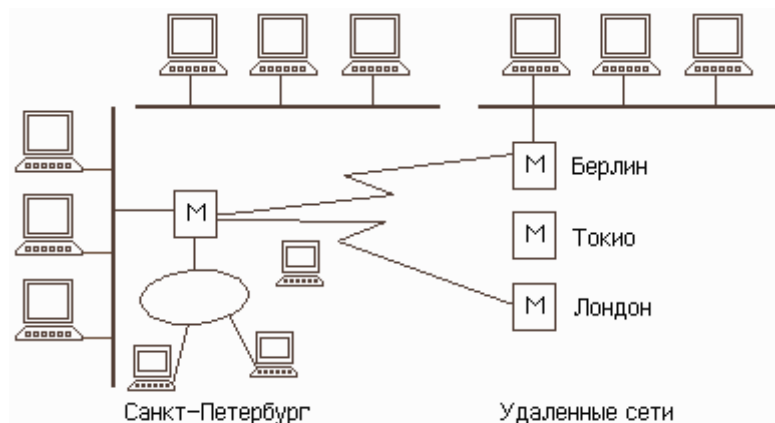
Для соединения локальных сетей используются следующие устройства, которые различаются между собой по назначению и возможностям:

- **Мост** (англ. Bridge) — связывает две локальные сети. **Передаёт данные между сетями в пакетном виде, не производя в них никаких изменений.** Ниже на рисунке показаны три локальные сети, соединённые двумя мостами.



Соединение локальных сетей посредством мостов

Здесь мосты создали **расширенную сеть**, которая обеспечивает своим пользователям доступ к **прежде недоступным ресурсам**. Кроме этого, мосты могут **фильтровать пакеты**, охраняя всю сеть от локальных потоков данных и пропуская наружу только те данные, которые предназначены для других сегментов сети.



- **Маршрутизатор** (англ. Router) объединяет сети с общим протоколом более эффективно, чем мост. Он позволяет, например, **расщеплять большие сообщения на более мелкие куски**, обеспечивая тем самым взаимодействие локальных сетей с разным размером пакета.

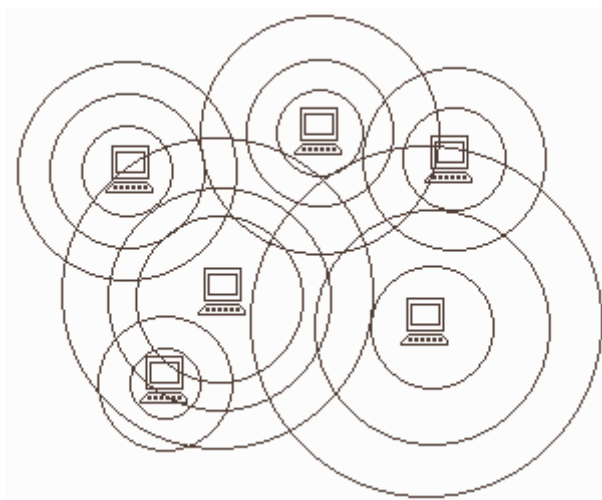
Маршрутизатор может пересылать пакеты на конкретный адрес (мосты только отфильтровывают ненужные пакеты), выбирать лучший путь для прохождения пакета и многое другое. Чем сложнее и больше сеть, тем больше выгода от использования маршрутизаторов.

- **Мостовой маршрутизатор** (англ. Brouter) — это гибрид моста и маршрутизатора, который сначала пытается выполнить маршрутизацию, где это только возможно, а затем, в случае неудачи, переходит в режим моста.

- **Шлюз** (англ. GateWay), в отличие от моста, применяется в случаях, когда соединяемые сети имеют **различные сетевые протоколы**. Поступившее в шлюз сообщение от одной сети преобразуется в другое сообщение, соответствующее требованиям следующей сети. Таким образом, шлюзы не просто соединяют сети, а позволяют им работать как единая сеть. С помощью шлюзов также локальные сети подсоединяются к **мэйнфреймам** — универсальным мощным компьютерам.

2.26. Как работают беспроводные сети?

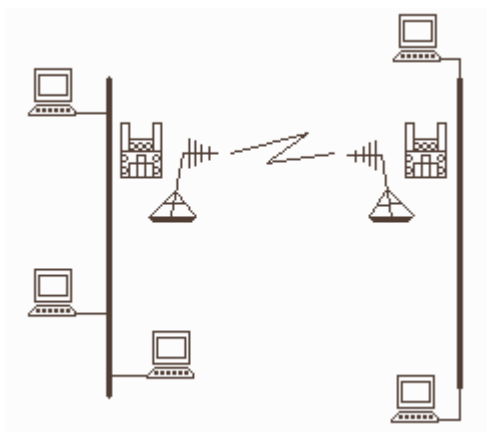
Беспроводные сети используются там, где прокладка кабелей затруднена, нецелесообразна или просто невозможна. Например, в исторических зданиях, промышленных помещениях с металлическим или железобетонным полом, в офисах, полученных в краткосрочную аренду, на складах, выставках, конференциях и т.п. [48]



Топология "Все-Со-Всеми"

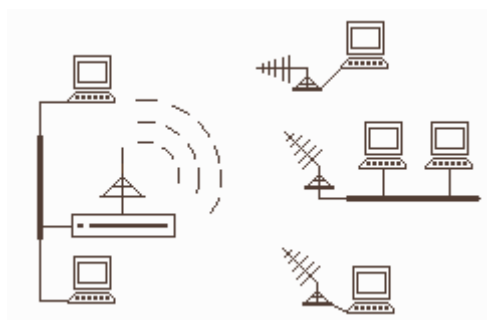
В этих случаях сеть реализуется при помощи **сетевых радио-адаптеров**, снабжённых **всенаправленными антеннами** и использующих в качестве среды передачи информации **радиоволны**. Такая сеть реализуется топологией **“Все-Со-Всеми”** и работоспособна при дальности 50–200 м.

Для связи между беспроводной и кабельной частями сети используется специальное устройство, называемое **точкой входа** (или **радиомостом**). Можно использовать и обычный компьютер, в котором установлены два сетевых адаптера — **беспроводной** и **кабельный**.



Топология "точка-точка"

Другой важной областью применения беспроводных сетей является **организация связи между удалёнными сегментами локальных сетей при отсутствии инфраструктуры передачи данных** (кабельных сетей общего доступа, высококачественных телефонных линий и др.), что типично для нашей страны. В этом случае для наведения беспроводных мостов между двумя удалёнными сегментами используются **радиомосты с антенной направленного типа**.



Топология типа "звезда"

Если в сеть нужно объединить **несколько сегментов**, то используется топология типа “звезда”. При этом в **центральном узле** устанавливается **всенаправленная антенна**, а **удалённых узлах — направленные**. Сети звездообразной топологии могут образовывать сети разнообразной конфигурации.

Сетевая магистраль с беспроводным доступом позволяет отказаться от использования медленных модемов.

2.27. Что такое сеть Интернет и как она работает?

Интернет — гигантская всемирная компьютерная сеть, объединяющая десятки тысяч сетей всего мира. Её назначение — обеспечить любому желающему постоянный доступ к любой информации. Интернет предлагает практически неограниченные информационные

ресурсы, полезные сведения, учёбу, развлечения, возможность общения с компетентными людьми, услуги удалённого доступа, передачи файлов, электронной почты и многое другое. Интернет обеспечивает принципиально новый способ общения людей, не имеющий аналогов в мире.

Благодаря сети стал доступен (бесплатно или за умеренную плату) огромный объём информации. Так, пользователь в любой стране может связаться с людьми, разделяющими его интересы, или получить ценные сведения в электронных библиотеках, даже если они находятся на другом конце света. Нужная информация окажется в его компьютере за считанные секунды, пройдя путь по длинной цепочке промежуточных компьютеров, по кабелям и по радио, через горы и моря, по дну океана и через спутник.

Интернет финансируется правительствами, научными и образовательными учреждениями, коммерческими структурами и миллионами частных лиц во всех частях света, но никто конкретно не является её владельцем. Управляет сетью "**Совет по архитектуре Интернет**", формируемый из приглашённых добровольцев.

Сеть была создана в 1984 году, и сейчас ею пользуются примерно сорок миллионов человек. Интернет всё время изменяется, поскольку имеет много квалифицированных пользователей, которые пишут программы для себя, а затем распространяют их среди желающих. Постоянно появляются новые серверы, а существующие обновляют свой "репертуар". Стремительно растут информационные потоки.

Как можно связаться с Интернет ?

Самый распространенный и недорогой способ — посредством модема и телефонной линии. При этом используются три типа подключения, отличающиеся друг от друга по объёму услуг и цене:

1. **почтовое** — позволяет только обмениваться электронной почтой с любым пользователем Интернет, самое дешёвое;
2. **сеансное** в режиме *on-line* ("на прямом проводе") — работа в диалоговом режиме — все возможности сети на время сеанса;
3. **прямое** (личное), самое дорогостоящее — все возможности в любое время.

При работе в сеансном режиме доступ к Интернет обычно покупается у **провайдеров** (англ. provide — предоставлять, обеспечивать) — фирм, предоставляющих доступ к некоторой части Интернет и поставляющих её пользователям разнообразные услуги.

Как связываются между собой сети в Интернет?

Отдельные участки Интернет представляют собой сети различной архитектуры, которые связываются между собой с помощью **маршрутизаторов**. Передаваемые данные разбиваются на небольшие порции, называемые **пакетами**. Каждый пакет перемещается по сети независимо от других пакетов. Сети в Интернет **неограниченно коммутируются (т.е. связываются) друг с другом**, потому что все компьютеры, участвующие в передаче данных, используют единый протокол коммуникации **ТСР/ІР** (читается "ти-си-пи / ай-пи"). На самом деле протокол ТСР/ІР — это два разных протокола, определяющих различные аспекты передачи данных в сети:

- **протокол TCP** (Transmission Control Protocol) — протокол управления передачей данных, использующий автоматическую повторную передачу пакетов, содержащих ошибки; этот протокол отвечает за разбиение передаваемой информации на пакеты и правильное восстановление информации из пакетов получателя;
- **протокол IP** (Internet Protocol) — протокол межсетевого взаимодействия, отвечающий за адресацию и позволяющий пакету на пути к конечному пункту назначения проходить по нескольким сетям.

Схема передачи информации по протоколу TCP/IP такова: протокол TCP разбивает информацию на пакеты и нумерует все пакеты; далее с помощью протокола IP все пакеты передаются получателю, где с помощью протокола TCP проверяется, все ли пакеты получены; после получения всех пакетов протокол TCP располагает их в нужном порядке и собирает в единое целое.

Каким образом пакет находит своего получателя ?

Каждый компьютер, подключенный к сети Интернет имеет два равноценных уникальных адреса: **цифровой IP-адрес и символический доменный адрес**. Присваивание адресов происходит по следующей схеме: международная организация Сетевой информационный центр выдает группы адресов владельцам локальных сетей, а последние распределяют конкретные адреса по своему усмотрению.

IP-адрес компьютера имеет длину 4 байта. Обычно первый и второй байты определяют **адрес сети**, третий байт определяет **адрес подсети**, а четвертый — **адрес компьютера в подсети**. Для удобства IP-адрес записывают в виде четырех чисел со значениями от 0 до 255, разделенных точками, например: 145.37.5.150. Адрес сети — 145.37; адрес подсети — 5; адрес компьютера в подсети — 150.

Доменный адрес (англ. domain — область), в отличие от цифрового, является символическим и легче запоминается человеком. Пример доменного адреса: **barsuk.les.nora.ru**. Здесь домен **barsuk** — имя реального компьютера, обладающего IP-адресом, домен **les** — имя группы, присвоившей имя этому компьютеру, домен **nora** — имя более крупной группы, присвоившей имя домену **les**, и т.д. В процессе передачи данных доменный адрес преобразуются в IP-адрес. Механизм такого преобразования подробно описан в [54].

2.28. Основные возможности, предоставляемые сетью Интернет

Интернет предоставляет своим пользователям разнообразные услуги и возможности (сервисы). Перечислим основные.

1. World Wide Web — главный информационный сервис.

World Wide Web (WWW, "Всемирная паутина") — гипертекстовая, а точнее, гипермедийная информационная система поиска ресурсов Интернет и доступа к ним.

Гипертекст — информационная структура, позволяющая устанавливать смысловые связи между элементами текста на экране компьютера таким образом, чтобы можно было

легко осуществлять переходы от одного элемента к другому. На практике в гипертексте некоторые слова выделяют путем подчёркивания или **окрашивания в другой цвет**. Выделение слова говорит о наличии связи этого слова с некоторым документом, в котором тема, связанная с выделенным словом, рассматривается более подробно.

Гипермедиа — это то, что получится, если в определении гипертекста заменить слово "текст" на "любые виды информации": звук, графику, видео. Такие гипермедийные ссылки возможны, поскольку наряду с текстовой информацией можно связывать и любую другую двоичную информацию, например, закодированный звук или графику. Так, если программа отображает карту мира и если пользователь выбирает на этой карте с помощью мыши какой-либо континент, программа может тут же дать о нём графическую, звуковую и текстовую информацию.

Система WWW построена на специальном протоколе передачи данных, который называется **протоколом передачи гипертекста HTTP** (читается "эйч-ти-ти-пи", HyperText Transfer Protocol). Всё содержимое системы WWW состоит из **WWW-страниц**.

WWW-страницы — гипермедийные документы системы World Wide Web. Создаются с помощью языка разметки гипертекста HTML (Hypertext markup language).

Язык HTML позволяет добавлять к текстовым документам специальные командные фрагменты — **тэги** (англ. **tag** — "этикетка, ярлык") таким образом, что становится возможным связывать с этими документами другие тексты, графику, звук и видео, задавать заголовки различных уровней, разделять текст на абзацы, строить таблицы и т.д. Например, заголовок документа может иметь такой вид:

<TITLE> Клуб любителей персиков </TITLE>

Одну WWW-страницу на самом деле обычно составляет **набор гипермедийных документов, расположенных на одном сервере, переплетённых взаимными ссылками и связанных по смыслу** (например, содержащих информацию об одном учебном заведении или об одном музее). Каждый документ страницы, в свою очередь, может содержать несколько экранных страниц текста и иллюстраций. Каждая WWW-страница имеет свой "титульный лист" (англ. "homepage") — гипермедийный документ, содержащий ссылки на главные составные части страницы. Адреса "титульных листов" распространяются в Интернет в качестве адресов страниц.

Личные страницы — такие WWW-страницы, которые принадлежат не фирмам и не организациям, а отдельным людям. Содержание и оформление такой страницы зависит только от её автора.

При работе с системой WWW пользователи имеют дело с **программами-клиентами системы, называемыми браузерами**.

Браузеры (англ. browse — листать, просматривать) — программы, с помощью которых пользователь организует диалог с системой WWW: просматривает WWW страницы, взаимодействует с WWW-серверами и другими ресурсами в Интернет.

Существуют сотни программ-браузеров. Самые популярные браузеры: **Netscape Navigator** и **Microsoft Internet Explorer**. Браузеры WWW умеют взаимодействовать с любыми типами серверов, используя при этом их собственные протоколы. Информацию, полученную от любого сервера, браузер WWW выводит на экран в стандартной, удобной для восприятия форме. При этом переключения с одного протокола на другой для пользователя часто остаются незамеченными.

2. Электронная почта.

Электронная почта (Electronic mail, англ. mail — почта, сокр. e-mail) служит для передачи текстовых сообщений в пределах Интернет, а также между другими сетями электронной почты. К тексту письма современные почтовые программы позволяют прикреплять звуковые и графические файлы, а также двоичные файлы — программы. При использовании электронной почты каждому абоненту присваивается уникальный почтовый адрес, формат которого имеет вид:
<имя пользователя> @ < имя почтового сервера>.
Например: earth@space.com, где earth — имя пользователя, space.com — имя компьютера, @ — разделительный символ "эт коммерческое".

Сообщения, поступающие по e-mail, хранятся в специальном "почтовом" компьютере в выделенной для получателя области дисковой памяти (его "почтовом ящике"), откуда он может их выгрузить и прочитать с помощью специальной программы-клиента. Для отсылки сообщения нужно знать электронный адрес абонента. При качественной связи электронное письмо доходит в любую точку мира в течение нескольких минут. Пользователи электронной почты стремятся придерживаться правил сетевого этикета (нэтикета), а для выражения эмоций используют схематические изображения человеческого лица, так называемые смайлики (англ. smiley, "улыбочка"), некоторые из которых приведены ниже.

Смайлики (рассматривайте, склонив голову влево)

:~)	улыбка	:-Q	курит	:~)	пускает
:~)))	хохот	:-@	кричит		слюнки
:~)	насморк	:-(грусть	8:-)	маленькая
:*)	пьяница	:!-(плачет		девочка
:~}	усатый	;~)	хитрец	:-*	съел горькое
:-[вампир	>:-(-	злится	:!-)	плачет от
:-)	панк	:0	зевает		счастья
				:-&	покаялся
					молчать
				O-)	аквалангист
				=8-)	носит очки

3. Система телеконференций Usenet (от Users Network).

Эта система организует коллективные обсуждения по различным направлениям, называемые телеконференциями. В каждой телеконференции проводится ряд дискуссий по конкретным темам. Сегодня Usenet имеет более десяти тысяч дискуссионных групп (NewsGroups) или телеконференций, каждая из которых посвящена определённой теме и является средством обмена мнениями. Телеконференции разбиты на несколько групп:

- **news** — вопросы, касающиеся системы телеконференций;
- **comp** — компьютеры и программное обеспечение;
- **rec** — развлечения, хобби и искусства;
- **sci** — научно-исследовательская деятельность и приложения;
- **soc** — социальные вопросы;
- **talk** — дебаты по различным спорным вопросам;
- **misc** — всё остальное.

Внутри этих категорий существует **иерархия**. Так, например, rec.music.beatles — это дискуссия о творчестве Битлз, входящая в подгруппу "музыка" группы дискуссий по искусству. Существует большой выбор **программ чтения телеконференций**, которые формируют материал дискуссий в упорядоченном виде и предоставляют в распоряжение корреспондентов. Аналог телеконференций в других сетях — "электронная доска объявлений" (Bulletin Board System, BBS).

4. Системы информационного поиска сети Интернет.

В Интернете представлена информация на любые темы, которые только можно себе представить. Но найти в ней нужную информацию не так-то легко из-за того, что сеть по своей природе не имеет чёткой структуры. Поэтому для ориентировки в Интернет и быстрого получения свежей справочной информации разработаны системы поиска информации. Все системы поиска информации Интернет располагаются на специально выделенных компьютерах с мощными каналами связи. Ежеминутно они бесплатно обслуживают огромное количество клиентов. Поисковые системы можно разбить на два типа:

- **предметные каталоги**, формируемые людьми-редакторами;
- **автоматические индексы**, формируемые специальными компьютерными программами, без участия людей.

Системы, основанные на предметных каталогах.

Используют базы данных, формируемые специалистами-редакторами, которые отбирают информацию, устанавливают связи для баз данных, организуют и снабжают данные в разных поисковых категориях перекрёстными ссылками. Кампании, владеющие предметными каталогами, непрерывно исследуют, описывают и каталогизируют содержимое WWW-серверов и других сетевых ресурсов, разбросанных по всему миру. В результате этой работы клиенты Интернет имеют постоянно обновляющиеся иерархические (древовидные) каталоги, на верхнем уровне которых собраны самые общие категории, такие как "бизнес", "наука", "искусство" и т.п., а элементы самого нижнего уровня представляют собой ссылки на отдельные WWW-страницы и серверы вместе с кратким описанием их содержимого.

Пример. Если нужно выяснить, какая в мире имеется информация о динозаврах, достаточно спуститься по иерархии:

Науки ==> Животные ==> Доисторические животные ==> Динозавры.



Каталоги, составленные людьми, более осмыслены, чем автоматические индексы. Их очень мало, так как их создание и поддержка требуют огромных затрат. Для примера рассмотрим самый популярный предметный каталог **Yahoo!**, который обладает

одной из крупнейших баз данных. Имеет информационные базы для детей и подростков. Поддерживает два основных метода работы с каталогом — поиск по ключевым словам и поиск по иерархическому дереву разделов. Не принимает запросов на естественном языке.

Автоматические индексы.

Переоценить их трудно. Поиск по ключевым словам в одной базе данных, занимающий в худшем случае несколько секунд, принесёт те же результаты, что и обшаривание всех WWW-страниц во всей сети Интернет.

Автоматический индекс состоит из трёх частей:

- **программы-робота;**
- **базы данных**, собираемой этим роботом;
- **интерфейса** для поиска в этой базе, с которым и работает пользователь.

Все эти компоненты функционируют без вмешательства человека. К автоматическим индексам следует прибегать только тогда, когда ключевые слова точно известны, например, фамилия человека или несколько специфических терминов из соответствующей области. Индексы получают информацию из каждого отдельного узла, регистрируют и индексируют её и добавляют к своим базам данных.



Среди известных индексов выделяется **AltaVista** — одна из самых мощных полностью автоматических поисковых систем. Обладает полнотекстовой базой данных. Выдаёт наибольшее количество ссылок. Проиндексировано 30 млн. страниц с 300 тысяч серверов и 4 млн. статей из телеконференций Usenet. За один день AltaVista обслуживает около 20 млн. запросов.

5. Программа пересылки файлов Ftp.

Перемещает копии файлов с одного узла Интернет на другой в соответствии с протоколом **FTP** (File Transfer Protocol — "протокол передачи файлов"). При этом не имеет значения, где эти узлы расположены и как соединены между собой. Компьютеры, на которых есть файлы для общего пользования, называются FTP-серверами. В Интернет имеется более 10 Терабайт бесплатных файлов и программ.

6. Программа удалённого доступа Telnet.

Позволяет входить в другую вычислительную систему, работающую в Интернет, с помощью протокола **TELNET**. Эта программа состоит из двух компонент: программы-клиента, которая выполняется на компьютере-клиенте, и программы-сервера, которая выполняется на компьютере-сервере.

Функции **программы-клиента**:

- **установление соединения с сервером;**

- приём от абонента входных данных, преобразование их к стандартному формату и отсылка серверу;
- приём от сервера результатов запроса в стандартном формате и переформатирование их в вид, удобный клиенту.

Функции программы-сервера:

- ожидание запроса в стандартной форме;
- обслуживание этого запроса;
- отсылка результатов программе-клиенту.

Telnet — простое и поэтому универсальное средство связи в Интернет.

В Интернет один и тот же узел сети может одновременно работать по нескольким протоколам. Поэтому крупные узлы сети сейчас обладают полным набором серверов, и к ним можно обращаться почти по любому из существующих протоколов.

2.29. Что такое мультимедиа и мультимедиа-компьютер?

Термин “мультимедиа” образован из слов “мульти” — много, и “медиа” — среда, носитель, средства сообщения, и в первом приближении его можно перевести как “многосредность” [47].

Мультимедиа — это собирательное понятие для различных компьютерных технологий, при которых используется несколько информационных сред, таких, как графика, текст, видео, фотография, движущиеся образы (анимация), звуковые эффекты, высококачественное звуковое сопровождение.

Мультимедиа-компьютер — это компьютер, снабженный аппаратными и программными средствами, реализующими технологию мультимедиа.

Области применения мультимедиа

- **Обучение с использованием компьютерных технологий** (Специальными исследованиями установлено, что из услышанного в памяти остается только четверть, из увиденного — треть, при комбинированном воздействии зрения и слуха — 50%, а если вовлечь учащегося в активные действия в процессе изучения при помощи мультимедийных приложений — 75% [46].
- **Информационная и рекламная служба.**
- **Развлечения, игры, системы виртуальной реальности.**

Технологию мультимедиа составляют две основные компоненты — аппаратная и программная.

Аппаратные средства мультимедиа

- **Основные** — компьютер с высокопроизводительным процессором, оперативной памятью 64 — 512 Мбайт, винчестерским накопителем ёмкостью 40 — 100 Гбайт и выше, накопителем на гибких магнитных дисках, манипуляторами, мультимедиа-монитором со встроенными стереодинамиками и видеоадаптером SVGA.

- **Специальные** — приводы CD-ROM; TV-тюнеры и фрейм-грабберы; графические акселераторы (ускорители), в том числе, для поддержки трёхмерной графики; платы видеовоспроизведения; устройства для ввода видеопоследовательностей; звуковые платы с установленными микшерами и музыкальными синтезаторами, воспроизводящими звучание реальных музыкальных инструментов; акустические системы с наушниками или динамиками и др.

Программные средства мультимедиа

- **Мультимедийные приложения** — энциклопедии, интерактивные курсы обучения по всевозможным предметам, игры и развлечения, работа с Интернет, тренажёры, средства торговой рекламы, электронные презентации, информационные киоски, установленные в общественных местах и предоставляющие различную информацию, и др.
- **Средства создания мультимедийных приложений** — редакторы видеоизображений; профессиональные графические редакторы; средства для записи, создания и редактирования звуковой информации, позволяющие подготавливать звуковые файлы для включения в программы, изменять амплитуду сигнала, наложить или убрать фон, вырезать или вставить блоки данных на каком-то временном отрезке; программы для манипуляции с сегментами изображений, изменения цвета, палитры; программы для реализации гипертекстов и др.

Технологии мультимедиа

- **Телевизионный приём** — вывод телевизионных сигналов на монитор компьютера на фоне работы других программ.
- **Видеозахват** — “захват” и “заморозка” в цифровом виде отдельных видеокадров.
- **Анимация** — воспроизведение последовательности картинок, создающее впечатление движущегося изображения.
- **Звуковые эффекты** — сохранение в цифровом виде звучания музыкальных инструментов, звуков природы или музыкальных фрагментов, созданных на компьютере, либо записанных и оцифрованных.
- **Трёхмерная (3D) графика** — графика, создаваемая с помощью изображений, имеющих не только длину и ширину, но и глубину.
- **Музыка MIDI** (Musical Instrument Digital Interface, цифровой интерфейс музыкальных инструментов) — стандарт, позволяющий подсоединять к компьютеру цифровые музыкальные инструменты, используемые при сочинении и записи музыки.
- **Виртуальная реальность** (Virtual Reality, VR). Слово “виртуальный” означает “действующий и проявляющий себя как настоящий”.

Виртуальная реальность — это высокоразвитая форма компьютерного моделирования, которая позволяет пользователю погрузиться в модельный мир и непосредственно действовать в нём. Зрительные, слуховые, осязательные и моторные ощущения пользователя при этом заменяются их имитацией, генерируемой компьютером.

Признаки устройств виртуальной реальности: **моделирование в реальном масштабе времени**; имитация окружающей обстановки с высокой степенью реализма; возможность воздействовать на окружающую обстановку и иметь при этом обратную связь.

Пример использования виртуальной реальности: архитектурно-строительная компания использует программное обеспечение, позволяющее заказчикам “посетить”

виртуальный образ будущего архитектурного сооружения задолго до того, как будет начато строительство.

2.30. Вопросы для самоконтроля

2.1. Какова роль аппаратуры (HardWare) и программного обеспечения (SoftWare) компьютера?

2.2. Какие основные классы компьютеров Вам известны?

2.3. В чём состоит принцип действия компьютеров?

2.4. Из каких простейших элементов состоит программа?

2.5. Что такое система команд компьютера?

2.6. Перечислите главные устройства компьютера.

2.7. Опишите функции памяти и функции процессора.

2.8. Назовите две основные части процессора. Каково их назначение?

2.9. Что такое регистры? Назовите некоторые важные регистры и опишите их функции.

2.10. Сформулируйте общие принципы построения компьютеров.

2.11. В чём заключается принцип программного управления? Как выполняются команды условных и безусловных переходов?

2.12. В чём суть принципа однородности памяти? Какие возможности он открывает?

2.13. В чём заключается принцип адресности?

2.14. Какие архитектуры называются "фон-неймановскими"?

2.15. Что такое команда? Что описывает команда?

2.16. Какого рода информацию может содержать адресная часть команды?

2.17. Приведите примеры команд одноадресных, двухадресных, трёхадресных.

2.18. Каким образом процессор при выполнении программы осуществляет выбор очередной команды?

2.19. Опишите основной цикл процесса обработки команд.

2.20. Что понимается под архитектурой компьютера? Какие характеристики компьютера определяются этим понятием? Верно ли, что общность архитектуры разных

компьютеров обеспечивает их совместимость в плане реализации функциональных элементов?

2.21. Что понимается под структурой компьютера? Какой уровень детализации описания компьютера может она обеспечить?

2.22. Перечислите распространённые компьютерные архитектуры.

2.23. Каковы отличительные особенности классической архитектуры?

2.24. Что собой представляет шина компьютера? Каковы функции общей шины (магистрالی)?

2.25. Какую функцию выполняют контроллеры?

2.26. Как характер решаемых задач связан с архитектурой компьютера?

2.27. Какие отличительные особенности присущи многопроцессорной архитектуре? Многомашинной архитектуре? Архитектуре с параллельным процессором?

2.28. Что такое центральный процессор?

2.29. Какие основные компоненты содержат в себе современные микропроцессоры?

2.30. Как конструктивно выполнены современные микропроцессоры?

2.31. Перечислите основные и производные единицы измерения количества памяти.

2.32. Назовите две основные разновидности памяти компьютера.

2.33. Перечислите основные компоненты внутренней памяти.

2.34. Что представляет собой ОЗУ? Каково её назначение?

2.35. В чём разница между памятью статической и динамической?

2.36. Что собой представляет модуль памяти типа SIMM? Какие другие типы модулей памяти Вы знаете?

2.37. Каково назначение кэш-памяти? Каким образом она реализуется?

2.38. Что такое специальная память? Характеризуйте её основные виды.

2.39. Что такое BIOS и какова её роль?

2.40. Каково назначение внешней памяти? Перечислите разновидности устройств внешней памяти.

2.41. Что собой представляет гибкий диск?

2.42. В чём суть магнитного кодирования двоичной информации?

- 2.43.** Как работают накопители на гибких магнитных дисках и накопители на жёстких магнитных дисках?
- 2.44.** Каковы достоинства и недостатки накопителей на компакт-дисках?
- 2.45.** Опишите работу стримера.
- 2.46.** Как работает аудиоадаптер? Видеоадаптер?
- 2.47.** Какие типы видеоплат используются в современных компьютерах?
- 2.48.** Назовите главные компоненты и основные управляющие клавиши клавиатуры.
- 2.49.** Перечислите основные компоненты видеосистемы компьютера.
- 2.50.** Как формируется изображение на экране цветного монитора?
- 2.51.** Как устроены жидкокристаллические мониторы? Проведите сравнение таких мониторов с мониторами, построенными на основе ЭЛТ.
- 2.52.** Опишите работу матричных, лазерных и струйных принтеров.
- 2.53.** Чем работа плоттера отличается от работы принтера?
- 2.54.** Опишите способ передачи информации посредством модема.
- 2.55.** Перечислите основные виды манипуляторов и опишите принципы их работы.
- 2.56.** Что понимают под персональным компьютером?
- 2.57.** Какие характеристики компьютера стандартизируются для реализации принципа открытой архитектуры?
- 2.58.** Что такое аппаратный интерфейс?
- 2.59.** Каково назначение контроллеров и адаптеров? В чём заключается разница между контроллером и адаптером?
- 2.60.** Что такое порты устройств? Охарактеризуйте основные виды портов.
- 2.61.** Перечислите основные блоки современного компьютера.
- 2.62.** Каково назначение межкомпьютерной связи?
- 2.63.** Опишите технологию "клиент–сервер".
- 2.64.** Каким образом преодолевается проблема несовместимости интерфейсов в компьютерных сетях?
- 2.65.** Что такое протокол коммуникации?
- 2.66.** Почему данные передаются при помощи пакетов?

- 2.67.** Охарактеризуйте основные виды сетевых топологий.
- 2.68.** Назовите характеристики распространённых сетевых архитектур.
- 2.69.** Дайте краткую характеристику специального сетевого оборудования.
- 2.70.** В каких областях и с какой целью применяются локальные сети?
- 2.71.** Перечислите основные сервисы сети Интернет.
- 2.72.** Что такое IP-адрес?
- 2.73.** Какие основные услуги предоставляет пользователям система WWW?
- 2.74.** Как организованы системы информационного поиска сети Интернет?
- 2.75.** Дайте сравнительную характеристику систем информационного поиска Yahoo! и Alta Vista.
- 2.76.** Охарактеризуйте основные виды технологий мультимедиа.
- 2.77.** Приведите примеры устройств "виртуальной реальности" и опишите принципы их работы.
- 2.78.** Каким вам представляется мультимедийный компьютер?
- 2.79.** Какой носитель информации используется для распространения мультимедиа-программ?
- 2.80.** Охарактеризуйте средства создания мультимедиа-приложений.
-
-

Глава 3. Классификация компьютеров

3.1. По каким критериям классифицируют компьютеры?

Существуют различные классификации компьютерной техники:

- по этапам развития (по поколениям);
- по архитектуре;
- по производительности;
- по условиям эксплуатации;
- по количеству процессоров;
- по потребительским свойствам и т.д.

Четких границ между классами компьютеров не существует. По мере совершенствования структур и технологии производства, появляются новые классы компьютеров, границы существующих классов существенно изменяются.

3.2. На чем основана классификация по поколениям?

Деление компьютерной техники на поколения — весьма условная, нестрогая классификация вычислительных систем по степени развития аппаратных и программных средств, а также способов общения с компьютером.

Идея делить машины на поколения вызвана к жизни тем, что за время короткой истории своего развития компьютерная техника проделала большую эволюцию как в смысле **элементной базы** (лампы, транзисторы, микросхемы и др.), так и в смысле **изменения её структуры, появления новых возможностей, расширения областей применения и характера использования.**

3.3. Краткая историческая справка

История счётных устройств насчитывает много веков. Ниже в хронологическом порядке приводятся некоторые наиболее значимые события этой истории, их даты и имена участников.

Около 500 г. н.э. Изобретение **счётов** (абака) — устройства, состоящего из набора костяшек, нанизанных на стержни.

1614 г. Шотландец **Джон Непер** изобрёл **логарифмы**. Вскоре после этого **Р. Биссакар** создал **логарифмическую линейку**.



Блез Паскаль

1642 г. Французский ученый **Блез Паскаль** приступил к созданию **арифметической машины** — механического устройства с шестернями, колёсами, зубчатыми рейками и т.п. Она умела "запоминать" числа и выполнять элементарные арифметические операции.



Перфокарта

1804 г. Французский инженер **Жаккар** изобрёл **перфокарты** для управления автоматическим ткацким станком, способным воспроизводить сложнейшие узоры. Работа станка программировалась колодой перфокарт, каждая из которых управляла одним ходом челнока.

1834 г. Английский ученый **Чарльз Бэббидж** составил проект "**аналитической**" **машины**, в которую входили: устройства ввода и вывода информации, запоминающее устройство для хранения чисел, устройство, способное выполнять арифметические операции, и устройство, управляющее последовательностью действий машины. Команды вводились с помощью перфокарт. Проект не был реализован.

1876 г. Английский инженер **Александр Белл** изобрёл **телефон**.

1890 г. Американский инженер **Герман Холлерит** создал **статистический табулятор**, в котором информация, нанесённая на перфокарты, расшифровывалась электрическим током. Табулятор использовался для обработки результатов переписи населения в США.

1892 г. Американский инженер **У. Барроуз** выпустил первый коммерческий **сумматор**.

1897 г. Английский физик **Дж. Томсон** сконструировал **электронно-лучевую трубку**.

1901 г. Итальянский физик **Гульельмо Маркони** установил **радиосвязь между Европой и Америкой**.

1904—1906 гг. Сконструированы электронные **диод** и **триод**.

1930 г. Профессор Массачусетского технологического института (МТИ) **Ванневар Буш** построил **дифференциальный анализатор**, с появлением которого связывают начало современной компьютерной эры.

Это была первая машина, способная решать сложные дифференциальные уравнения, которые позволяли предсказывать поведение таких движущихся объектов, как самолет, или действие силовых полей, например, гравитационного поля.



Алан Тьюринг

1936 г. Английский математик [Алан Тьюринг](#) и независимо от него Э. Пост выдвинули и разработали **концепцию абстрактной вычислительной машины**. Они доказали принципиальную возможность решения автоматами любой проблемы при условии возможности её алгоритмизации.

1938 г. Немецкий инженер **Конрад Цузе** построил первый чисто **механический компьютер**.



Конрад Цузе

1938 г. Американский математик и инженер [Клод Шеннон](#) показал **возможность применения аппарата математической логики для синтеза и анализа релейно-контактных переключательных схем.**

1939 г. Американец болгарского происхождения профессор физики **Джон Атанасофф** создал прототип вычислительной машины на базе двоичных элементов.



Компьютер "Марк—1"

1941 г. **Конрад Цузе** сконструировал первый универсальный компьютер на электромеханических элементах. Он работал с двоичными числами и использовал представление чисел с плавающей запятой.

1944 г. Под руководством американского математика **Говарда Айкена** создана автоматическая вычислительная машина "**Марк—1**" с программным управлением. Она была построена на электромеханических реле, а программа обработки данных вводилась с перфоленты.



Джон фон Нейман

1945 г. **Джон фон Нейман** в отчёте "Предварительный доклад о машине Эдвак" сформулировал **основные принципы работы и компоненты современных компьютеров.**



Компьютер "Эниак", 1946 г.

1946 г. Американцы **Дж. Эккерт** и **Дж. Моучли** сконструировали первый электронный цифровой компьютер "**Эниак**" (Electronic Numerical Integrator and Computer). Машина имела 20 тысяч электронных ламп и 1,5 тысячи реле. Она работала в тысячу раз быстрее, чем "Марк—1", выполняя за одну секунду 300 умножений или 5000 сложений.



Транзистор

1948 г. В американской фирме Bell Laboratories физики **Уильям Шокли**, **Уолтер Браттейн** и **Джон Бардин** создали **транзистор**. За это достижение им была присуждена Нобелевская премия.

1948 г. **Норберт Винер** (Norbert Wiener) опубликовал книгу "**Кибернетика**", оказавшую влияние на все последующие исследования в области искусственного интеллекта.

1949 г. В Англии под руководством **Мориса Уилкса** построен первый в мире компьютер с хранимой в памяти программой **EDSAC**.

1951 г. В Киеве построен первый в континентальной Европе компьютер **МЭСМ** (малая электронная счетная машина), имеющий 600 электронных ламп. Создатель **С.А. Лебедев**.

1951—1955 гг. Благодаря деятельности российских ученых **С.А. Лебедева**, **М.В. Келдыша**, **М.А. Лаврентьева**, **И.С. Брука**, **М.А. Карцева**, **Б.И. Рамеева**, **В.С. Антонова**, **А.Н. Невского**, **Б.И. Буркова** и руководимых ими коллективов Советский Союз вырвался в число лидеров вычислительной техники, что позволило в короткие сроки решить важные научно-технические задачи овладения ядерной энергией и исследования Космоса.

1952 г. Под руководством **С.А. Лебедева** в Москве построен компьютер **БЭСМ—1** (большая электронная счетная машина) — на то время самая производительная машина в Европе и одна из лучших в мире.

1953 г. **Джей Форрестер** реализовал **оперативную память на магнитных сердечниках** (core memory), которая существенно удешевила компьютеры и увеличила их быстродействие. Память на магнитных сердечниках широко использовалась до начала 70-х годов. На смену ей пришла память на полупроводниковых элементах.

1955—1959 гг. Российские ученые **А.А. Ляпунов**, **С.С. Камынин**, **Э.З. Любимский**, **А.П. Ершов**, **Л.Н. Королев**, **В.М. Курочкин**, **М.Р. Шура-Бура** и др. создали "**программирующие программы**" — прообразы трансляторов. **В.В. Мартынюк** создал **систему символического кодирования** — средство ускорения разработки и отладки программ.

1955—1959 гг. Заложены фундамент теории программирования (**А.А. Ляпунов**, **Ю.И. Янов**, **А.А. Марков**, **Л.А. Калужин**) и численных методов (**В.М. Глушков**, **А.А. Самарский**, **А.Н. Тихонов**). Моделируются схемы механизма мышления и процессов генетики, алгоритмы диагностики медицинских заболеваний (**А.А. Ляпунов**, **Б.В. Гнеденко**, **Н.М. Амосов**, **А.Г. Ивахненко**, **В.А. Ковалевский** и др.).



Джон Бэкус



Интегральная схема

1958 г. Джек Килби из фирмы Texas Instruments создал **первую интегральную схему**.

1957 г. Первое сообщение о языке **Фортран** (Джон Бэкус).

1957 г. Американской фирмой NCR создан **первый компьютер на транзисторах**.

1959 г. Под руководством **С.А. Лебедева** создана машина **БЭСМ—2** производительностью 10 тыс. опер./с. С ее применением связаны расчеты запусков космических ракет и первых в мире искусственных спутников Земли.



С.А. Лебедев

1959 г. Создана машина **М—20**, главный конструктор **С.А. Лебедев**. Для своего времени одна из самых быстродействующих в мире (20 тыс. опер./с.). На этой машине было решено большинство теоретических и прикладных задач, связанных с развитием самых передовых областей науки и техники того времени. На основе М—20 была создана уникальная многопроцессорная **М—40** — самая быстродействующая ЭВМ того времени в мире (40 тыс. опер./с.). На смену М—20 пришли полупроводниковые **БЭСМ—4** и **М—220** (200 тыс. опер./с.).

1959 г. Первое сообщение о языке **Алгол**, который надолго стал стандартом в области языков программирования.

1961 г. Фирма IBM Deutschland реализовала **подключение компьютера к телефонной линии с помощью модема**.

1964 г. Начат выпуск семейства машин третьего поколения — IBM/360.

1965 г. Дж. Кемени и Т. Курц в Дортмундском колледже (США) разработали язык программирования **Бейсик**.



БЭСМ—6

1965 г. Сеймур Пейперт (Seymour Papert) разработал язык **LOGO** — компьютерный язык для детей.

1967 г. Под руководством С.А. Лебедева организован крупно-серийный выпуск шедевра отечественной вычислительной техники — миллионника **БЭСМ—6**, — самой быстродействующей машины в мире. За ним последовал "**Эльбрус**" — ЭВМ нового типа, производительностью 10 млн. опер./с.



Никлаус Вирт

1968 г. Основана фирма **Intel**, впоследствии ставшая признанным лидером в области производства микропроцессоров и других компьютерных интегральных схем.

1970 г. Швейцарец **Никлаус Вирт** разработал язык Паскаль.

1971 г. **Эдвард Хофф** разработал микропроцессор **Intel—4004**, состоящий из 2250 транзисторов, размещённых в кристалле размером не больше шляпки гвоздя. Этот микропроцессор стал поистине революционным изобретением, открывшем путь к созданию искусственных интеллектуальных систем вообще и персонального компьютера в частности.

1971 г. Французский учёный **Алан Колмари** разработал язык логического программирования **Пролог** (PROgramming in LOGic).



Деннис Ритчи

1972 г. Деннис Ритчи из Bell Laboratories разработал [язык Си](#).

1973 г. Кен Томпсон и Деннис Ритчи создали операционную систему [UNIX](#).

1973 г. Фирма IBM (International Business Machines Corporation) сконструировала первый жёсткий диск типа ["винчестер"](#).

1974 г. Фирма Intel разработала **первый универсальный восьмиразрядный микропроцессор 8080** с 4500 транзисторами.



Альтаир

1974 г. Эдвард Робертс, молодой офицер ВВС США, инженер-электронщик, построил на базе процессора 8080 микрокомпьютер **Альтаир**, имевший огромный коммерческий успех, продававшийся по почте и широко использовавшийся для домашнего применения. Компьютер назван по имени звезды, к которой был запущен межпланетный корабль "Энтерпрайз" из телесериала "Космическая одиссея".



Билл Гейтс
и Пол Аллен

1975 г. Молодой программист **Пол Аллен** и студент Гарвардского университета **Билл Гейтс** реализовали для **Альтаира** язык **Бейсик**. Впоследствии они основали фирму **Майкрософт** (Microsoft), являющуюся сегодня крупнейшим производителем программного обеспечения.

1975 г. Фирма IBM начала продажу [лазерных принтеров](#).



Apple—1

1976 г. Студенты **Стив Возняк** и **Стив Джобс**, устроив мастерскую в гараже, реализовали компьютер **Apple—1**, положив начало корпорации [Apple](#).



Стивен Джобс и
Стефан Возняк

1978 г. Фирма **Intel** выпустила **микропроцессор 8086**.

1979 г. Фирма **Intel** выпустила **микропроцессор 8088**. Корпорация IBM приобрела крупную партию этих процессоров для вновь образованного подразделения по разработке и производству персональных компьютеров.

1979 г. Фирма **SoftWare Arts** разработала первый пакет деловых программ **VisiCalc** (Visible Calculator) для персональных компьютеров.

1980 г. Корпорация **Control Data** выпустила [суперкомпьютер Cyber \(Сайбер\) 205](#).

1980 г. Японские компании **Sharp, Sanyo, Panasonic, Casio** и американская фирма **Tandy** вынесли на рынок первый **карманный компьютер**, обладающий всеми основными свойствами больших компьютеров.

1981 г. Фирма **IBM** выпустила первый **персональный компьютер IBM PC** на базе микропроцессора 8088.

1982 г. Фирма **Intel** выпустила **микропроцессор 80286**, содержащий 134 000 транзисторов и способный выполнять любые программы, написанные для его предшественников. С тех пор такая программная совместимость остается отличительным признаком семейства микропроцессоров Intel.

1982 г. **Митч Капор** (Mitch Kapor) представил систему **Lotus 1—2—3**, которая победила в конкурентной борьбе Visicalc.



Lisa

1983 г. Корпорация **Apple Computers** построила персональный компьютер **Lisa** — первый офисный компьютер, управляемый манипулятором мышь.

1983 г. Гибкие диски получили распространение в качестве стандартных носителей информации.



Андерс Хейльсберг

1983 г. Фирмой **Borland** выпущен в продажу компилятор Turbo Pascal, разработанный **Андерсом Хейльсбергом** (Anders Hejlsberg).

1984 г. Создан первый компьютер типа Laptop (наколенный), в котором системный блок объединен с дисплеем и клавиатурой в единый блок.

1984 г. Фирмы **Sony** и **Phillips** разработали стандарт записи компакт-дисков CD-ROM.



Macintosh

1984 г. Корпорация **Apple Computer** выпустила компьютер **Macintosh** на 32-разрядном процессоре **Motorola 68000** — первую модель знаменитого впоследствии семейства Macintosh с удобной для пользователя операционной системой, развитыми графическими возможностями, намного превосходящими в то время те, которыми обладали стандартные IBM-совместимые ПК с MS-DOS. Эти компьютеры быстро приобрели миллионы поклонников и стали вычислительной платформой для целых отраслей, таких например, как издательское дело и образование.

1984 г. Появилась некоммерческая компьютерная сеть **FIDO**. Ее создатели **Том Дженнингс** и **Джон Мэдил**. В 1995 году в мире насчитывалось около 20 тысяч узлов этой сети, объединяющих 3 млн. человек.

1985 г. Фирма **Intel** выпустила **микропроцессор 80386**, насчитывающий уже 275000 транзисторов. Этот 32-разрядный "многозадачный" процессор обеспечивал возможность одновременного выполнения нескольких программ.

1985 г. **Бьярн Страуструп** из **Bell Laboratories** опубликовал описание созданного им объектно-ориентированного языка **C++**.

1989 г. Американская фирма **Poquet Computers Corporation** представила новый компьютер класса **Subnotebook** — **Pocket PC**.

1989 г. **Тим Бернерс-Ли** предложил **язык гипертекстовой разметки HTML** (HyperText Markup Language) в качестве одного из компонентов технологии разработки распределенной гипертекстовой системы **World Wide Web**.

1989 г. Фирма **Intel** выпустила микропроцессор **Intel 486 DX**. Поколение процессоров i486 ознаменовало переход от работы на компьютере через командную строку к режиму "укажи и щелкни". Intel 486 стал первым микропроцессором со встроенным математическим сопроцессором, который существенно ускорил обработку данных, выполняя сложные математические действия вместо центрального процессора. Количество транзисторов — 1,2 млн.

Корпорация Microsoft выпустила графическую оболочку [MS Windows 3.0](#).

1990 г. Выпуск и ввод в эксплуатацию векторно-конвейерной суперЭВМ "**Эльбрус 3.1**". Разработчики — **Г.Г. Рябов, А.А. Соколов, А.Ю. Бяков**. Производительность в однопроцессорном варианте — 400 мегафлопов.



Линус Торвальдс

1991 г. Финский студент **Линус Торвальдс** (Linus Torvalds) распространил среди пользователей Интернет первый прототип своей операционной системы [Linux](#). Заинтересованные в этой работе программисты стали поддерживать Linux, добавляя драйверы устройств, разрабатывая разные продвинутые приложения и др. Атмосфера

работы энтузиастов над полезным проектом, а также свободное распространение и использование исходных текстов стали основой феномена Linux. В настоящее время Linux — очень мощная система, к тому же — бесплатная.

1992 г. В этом году начался бурный рост популярности Internet и World Wide Web в связи с появлением **web-браузера Mosaic**, разработанного в Национальном центре по приложениям для суперкомпьютеров в Университете штата Иллинойс. Разработчики **Эрик Бина** и **Марк Андрессен**.

1993 г. Фирма **Intel** выпустила микропроцессор **Pentium**, который научил компьютеры работать с атрибутами "реального мира" — такими, как звук, голосовая и письменная речь, фотоизображения.

1994 г. Начало выпуска фирмой **Power Mac** серии фирмы Apple Computers — **Power PC**.

1994 г. Компания **Netscape Communication** выпустила браузер **Netscape Navigator**.

1995 г. Фирма **Microsoft** выпустила в свет операционную систему **[Windows 95](#)**.

1995 г. Фирма **Microsoft** выпустила браузер **Internet Explorer**. Началась война браузеров, в которой пока побеждает Internet Explorer.

1995 г. Фирма **Intel** выпустила микропроцессор **Pentium Pro**, насчитывающий 5,5 миллионов транзисторов. Процессор разрабатывался как мощное средство наращивания быстродействия 32-разрядных приложений для серверов и рабочих станций, систем автоматизированного проектирования, программных пакетов, используемых в машиностроении и научной работе. Все процессоры Pentium Pro оснащены второй микросхемой кэш-памяти, еще больше увеличивающей быстродействие.

1997 г. Фирма **Intel** выпустила микропроцессор **Pentium II**, насчитывающий 7,5 миллионов транзисторов. Процессор Pentium II использует технологию Intel MMX, обеспечивающую эффективную обработку аудио, визуальных и графических данных. Кристалл и микросхема высокоскоростной кэш-памяти помещены в корпус с односторонним контактом, который устанавливается на системной плате с помощью одностороннего разъема — в отличие от прежних процессоров, имевших множество контактов. Процессор дает пользователям возможность вводить в компьютер и обрабатывать цифровые фотоизображения, создавать и редактировать тексты, музыкальные произведения, сценки для домашнего кино, передавать видеоизображения по обычным телефонным линиям.

1997 г. Компания **Sun Microsystems** приняла стандарт объектно-ориентированного языка программирования **Java** (произносится "джава"), созданного для реализации принципа "Написано однажды — работает везде". В применении к интернету Java — технология создания "апплетов" — небольших программ, которые загружаются на компьютер пользователя вместе со страницей сайта и позволяют "оживлять" эту страницу. Апплеты могут обеспечивать странице дополнительную функциональность, например, реализовывать мультипликационные иллюстрации.

1998 г. Выпуск в свет операционной системы **[Windows 98](#)**.

1999 г. Появление 64-разрядного микропроцессора **Mersed**.

2000 г. Появление 64-разрядных микропроцессоров **Itanium** и **AMD**.

2000 г. Выпуск в свет операционной системы [Windows 2000](#).

3.4. Какие компьютеры относятся в первому поколению?

К **первому поколению** обычно относят машины, созданные на рубеже 50-х годов. В их схемах использовались **электронные лампы**. Эти компьютеры были **огромными, неудобными и слишком дорогими** машинами, которые могли приобрести только крупные корпорации и правительства. Лампы потребляли огромное количество электроэнергии и выделяли много тепла.



Электронная
лампа



Компьютер "Эниак".
Первое поколение

Набор команд был небольшой, схема арифметико-логического устройства и устройства управления достаточно проста, программное обеспечение практически отсутствовало. Показатели объема оперативной памяти и быстродействия были низкими. Для ввода-вывода использовались перфоленды, перфокарты, магнитные ленты и печатающие устройства.

Быстродействие порядка 10-20 тысяч операций в секунду.

Но это только техническая сторона. Очень важна и другая — способы использования компьютеров, стиль программирования, особенности математического обеспечения.



Перфокарта

Программы для этих машин писались **на языке конкретной машины**. Математик, составивший программу, садился за пульт управления машины, вводил и отлаживал программы и производил по ним счет. Процесс отладки был наиболее длительным по времени.

Несмотря на ограниченность возможностей, эти машины позволили выполнить сложнейшие расчёты, необходимые для прогнозирования погоды, решения задач атомной энергетики и др.

Опыт использования машин первого поколения показал, что существует огромный разрыв между временем, затрачиваемым на разработку программ, и временем счета.



ЭВМ "Урал"

Эти проблемы начали преодолевать путем интенсивной **разработки средств автоматизации программирования, создания систем обслуживающих программ, упрощающих работу на машине и увеличивающих эффективность её использования**. Это, в свою очередь, потребовало значительных изменений в структуре компьютеров, направленных на то, чтобы приблизить её к требованиям, возникшим из опыта эксплуатации компьютеров.

Отечественные машины первого поколения: МЭСМ (малая электронная счётная машина), БЭСМ, Стрела, Урал, М—20.

[Здесь](#) вы найдете богатый материал по истории отечественной вычислительной техники.

3.5. Какие компьютеры относятся ко второму поколению?

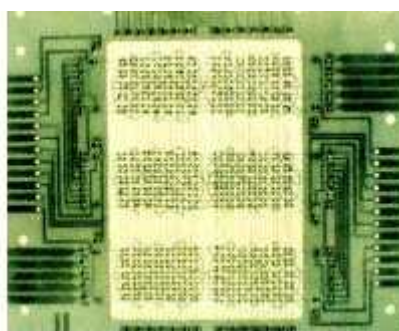


БЭСМ—6. Второе поколение



Транзистор

Второе поколение компьютерной техники — машины, сконструированные примерно в 1955—65 гг. Характеризуются использованием в них как **электронных ламп**, так и **дискретных транзисторных логических элементов**. Их оперативная память была построена на магнитных сердечниках. В это время стал расширяться диапазон применяемого оборудования ввода-вывода, появились высокопроизводительные устройства для работы с магнитными лентами, магнитные барабаны и первые магнитные диски.



Память на магнитных сердечниках

Быстродействие — до сотен тысяч операций в секунду, **ёмкость памяти** — до нескольких десятков тысяч слов.

Появились так называемые **языки высокого уровня**, средства которых допускают описание всей необходимой последовательности вычислительных действий **в наглядном, легко воспринимаемом виде**.

Программа, написанная на алгоритмическом языке, непонятна компьютеру, воспринимающему только язык своих собственных команд. Поэтому специальные программы, которые называются **трансляторами**, переводят программу с языка высокого уровня на машинный язык.

Появился широкий набор библиотечных программ для решения разнообразных математических задач. Появились **мониторные системы**, управляющие режимом трансляции и исполнения программ. Из мониторных систем в дальнейшем выросли современные операционные системы.

Операционная система — важнейшая часть программного обеспечения компьютера, предназначенная для автоматизации планирования и организации процесса обработки программ, ввода-вывода и управления данными, распределения ресурсов, подготовки и отладки программ, других вспомогательных операций обслуживания.

Таким образом, операционная система является программным расширением устройства управления компьютера.

Для некоторых машин второго поколения уже были созданы операционные системы с ограниченными возможностями.

Машинам второго поколения была свойственна **программная несовместимость**, которая затрудняла организацию крупных информационных систем. Поэтому в середине 60-х годов наметился переход к созданию компьютеров, программно совместимых и построенных на микроэлектронной технологической базе.

[Здесь](#) вы найдете описания отечественных компьютеров второго поколения.

3.6. В чем особенности компьютеров третьего поколения?



Компьютер IBM—360.
Третье поколение

Машины третьего поколения созданы примерно после 60-х годов. Поскольку процесс создания компьютерной техники шел непрерывно, и в нём участвовало множество людей из разных стран, имеющих дело с решением различных проблем, трудно и бесполезно пытаться установить, когда "поколение" начиналось и заканчивалось. Возможно, наиболее важным критерием различия машин второго и третьего поколений является критерий, основанный на понятии архитектуры.



Интегральная схема

Машины третьего поколения — это семейства машин с единой архитектурой, т.е. программно совместимых. В качестве элементной базы в них используются интегральные схемы, которые также называются микросхемами.

Машины третьего поколения имеют развитые операционные системы. Они обладают возможностями мультипрограммирования, т.е. одновременного выполнения нескольких программ. Многие задачи управления памятью, устройствами и ресурсами стала брать на себя операционная система или же непосредственно сама машина.

Примеры машин третьего поколения — семейства IBM—360, IBM—370, ЕС ЭВМ (Единая система ЭВМ), СМ ЭВМ (Семейство малых ЭВМ) и др.

Быстродействие машин внутри семейства изменяется от нескольких десятков тысяч до миллионов операций в секунду. Ёмкость оперативной памяти достигает нескольких сотен тысяч слов.

Краткое описание процесса изготовления микросхем

1. Разработчики с помощью компьютера создают электрическую схему новой микросхемы. Для этого они вводят в компьютер перечень свойств, которыми должна обладать микросхема, а компьютер с помощью специальной программы разрабатывает детальную структуру соединений и конструкций всех взаимодействующих элементов микросхемы.
2. Компьютер создаёт схемы расположения элементов на поверхности полупроводникового кристалла кремния. По этим схемам изготавливаются фотошаблоны — стеклянные пластинки со штриховым рисунком. Через фотошаблоны специальными лампами или источниками рентгеновского излучения, а иногда, и электронными пучками, освещают (засвечивают) нанесённый на поверхность кристалла кремния слой фото- или, соответственно, рентгеночувствительного лака.
3. Засвеченные (или, наоборот, незасвеченные) участки лака меняют свои свойства и удаляются специальными растворителями. Этот процесс называется травлением. Вместе с лаком с поверхности кристалла кремния удаляется и слой окисла, и эти места становятся доступными для легирования — внедрения в кристаллическую решётку кремния атомов бора или фосфора. Легирование обычно требует нагрева пластинки в парах нужного элемента до 1100 — 1200 °С.
4. Последовательно меняя шаблоны и повторяя процедуры травления и легирования, создают один за другим слои будущей микросхемы. При этом на одной пластинке кристалла кремния создаётся множество одинаковых микросхем.
5. Каждая микросхема проверяется на работоспособность. Негодные выбраковываются.
6. После завершения всех операций пластинки нарезаются на отдельные кристаллики с микросхемами, к ним присоединяют выводы и устанавливают в корпуса.

[Здесь](#) вы найдете описания отечественных компьютеров третьего поколения.

3.7. Что характерно для машин четвёртого поколения?

Четвёртое поколение — это теперешнее поколение компьютерной техники, разработанное после 1970 года.

Наиболее важный в концептуальном отношении критерий, по которому эти компьютеры можно отделить от машин третьего поколения, состоит в том, что машины четвёртого поколения проектировались в расчете на эффективное использование современных [высокоуровневых языков](#) и упрощение процесса программирования для конечного пользователя.

В аппаратном отношении для них характерно широкое использование **интегральных схем** в качестве элементной базы, а также наличие быстродействующих запоминающих устройств с произвольной выборкой ёмкостью в десятки мегабайт.

С точки зрения структуры машины этого поколения представляют собой **многопроцессорные и многомашинные комплексы**, работающие на общую память и общее поле внешних устройств. Быстродействие составляет до нескольких десятков миллионов операций в секунду, ёмкость оперативной памяти порядка 1 — 64 Мбайт.

Для них характерны:

- применение [персональных компьютеров](#);
- телекоммуникационная обработка данных;
- [компьютерные сети](#);
- широкое применение [систем управления базами данных](#);
- элементы интеллектуального поведения систем обработки данных и устройств.

3.8. Какими должны быть компьютеры пятого поколения?

Разработка последующих поколений компьютеров производится на основе **больших интегральных схем повышенной степени интеграции**, использования оптоэлектронных принципов (**лазеры, голография**).

Развитие идет также по пути "**интеллектуализации**" компьютеров, устранения барьера между человеком и компьютером. Компьютеры будут способны воспринимать информацию с рукописного или печатного текста, с бланков, с человеческого голоса, узнавать пользователя по голосу, осуществлять перевод с одного языка на другой.

В компьютерах пятого поколения произойдёт качественный переход от обработки данных к обработке знаний.

Архитектура компьютеров будущего поколения будет содержать два основных блока. Один из них — это **традиционный** компьютер. Но теперь он лишён связи с пользователем. Эту связь осуществляет блок, называемый термином "**интеллектуальный интерфейс**". Его задача — понять текст, написанный на естественном языке и содержащий условие задачи, и перевести его в работающую программу для компьютера.

Будет также решаться проблема децентрализации вычислений с помощью компьютерных сетей, как больших, находящихся на значительном расстоянии друг от друга, так и миниатюрных компьютеров, размещённых на одном кристалле полупроводника.

3.9. На какие типы делятся компьютеры по условиям эксплуатации?

По условиям эксплуатации компьютеры делятся на два типа:

- офисные (универсальные);
- специальные.

Офисные предназначены для решения широкого класса задач при нормальных условиях эксплуатации.



Индустриальный компьютер

Специальные компьютеры служат для решения более узкого класса задач или даже одной задачи, требующей многократного решения, и функционируют в особых условиях эксплуатации. Машинные ресурсы специальных компьютеров часто ограничены. Однако их узкая ориентация позволяет реализовать заданный класс задач наиболее эффективно.

Специальные компьютеры управляют технологическими установками, работают в операционных или машинах скорой помощи, на ракетах, самолётах и вертолётах, вблизи высоковольтных линий передач или в зоне действия радаров, радиопередатчиков, в неотопляемых помещениях, под водой на глубине, в условиях пыли, грязи, вибраций, взрывоопасных газов и т.п. Существует много моделей таких компьютеров. Познакомимся с одной из них.



Ergotouch

Компьютер **Ergotouch** (Эрготач) [48] исполнен в литом алюминиевом полностью герметичном корпусе, который легко открывается для обслуживания. Стенки компьютера поглощают практически все электромагнитные излучения как изнутри, так и снаружи. Машина оборудована экраном, чувствительным к прикосновениям. Компьютер можно, не выключая, мыть из шланга, дезинфицировать, дезактивировать, обезжиривать. Высочайшая надежность позволяет использовать его как средство управления и контроля технологическими процессами в реальном времени. Компьютер легко входит в локальную

сеть предприятия.

Важное направление в создании промышленных компьютеров — разработка "**операторского интерфейса**" — пультов управления, дисплеев, клавиатур и указательных устройств во всевозможных исполнениях. От этих изделий напрямую зависит комфортность и результативность труда операторов.

3.10. На какие типы делятся компьютеры по производительности и характеру использования?

По производительности и характеру использования компьютеры можно условно подразделить на:

- микрокомпьютеры, в том числе — персональные компьютеры;
- миникомпьютеры;
- мэйнфреймы (универсальные компьютеры);
- суперкомпьютеры.

Микрокомпьютеры — это компьютеры, в которых центральный процессор выполнен в виде микропроцессора.

Продвинутые модели микрокомпьютеров имеют несколько микропроцессоров. Производительность компьютера определяется не только характеристиками применяемого микропроцессора, но и ёмкостью оперативной памяти, типами периферийных устройств, качеством конструктивных решений и др.

Микрокомпьютеры представляют собой инструменты для решения разнообразных сложных задач. Их микропроцессоры с каждым годом увеличивают мощность, а периферийные устройства — эффективность. Быстродействие — порядка 1 — 10 миллионов операций в сек.

Разновидность микрокомпьютера — **микроконтроллер**. Это основанное на микропроцессоре специализированное устройство, встраиваемое в систему управления или технологическую линию.

Персональные компьютеры (ПК) — это микрокомпьютеры универсального назначения, рассчитанные на одного пользователя и управляемые одним человеком.

В класс персональных компьютеров входят различные машины — от **дешёвых** домашних и игровых с небольшой оперативной памятью, с памятью программы на кассетной ленте и обычным телевизором в качестве дисплея (80-е годы), до **сверхсложных** машин с мощным процессором, винчестерским накопителем ёмкостью в десятки Гигабайт, с цветными графическими устройствами высокого разрешения, средствами мультимедиа и другими дополнительными устройствами.

Персональный компьютер должен удовлетворять следующим требованиям:

- стоимость от нескольких сотен до 5 —10 тыс. долларов;
- наличие внешних ЗУ на магнитных дисках;
- объём оперативной памяти не менее 32 Мбайт;
- наличие операционной системы;
- способность работать с программами на языках высокого уровня;
- ориентация на пользователя-непрофессионала (в простых моделях).

Миникомпьютерами и **суперминикомпьютерами** называются машины, конструктивно выполненные в одной стойке, т.е. занимающие объём порядка половины кубометра. Сейчас компьютеры этого класса вымирают, уступая место микрокомпьютерам.

Мэйнфреймы предназначены для решения широкого класса научно-технических задач и являются сложными и дорогими машинами. Их целесообразно применять в больших системах при наличии не менее 200 — 300 рабочих мест.

Централизованная обработка данных на мэйнфрейме обходится примерно в 5 — 6 раз дешевле, чем распределённая обработка при клиент-серверном подходе.

Известный мэйнфрейм **S/390** фирмы IBM обычно оснащается не менее чем тремя процессорами. Максимальный объём оперативного хранения достигает 342 Терабайт.

Производительность его процессоров, пропускная способность каналов, объём оперативного хранения позволяют наращивать число рабочих мест в диапазоне от 20 до 200000 с помощью простого добавления процессорных плат, модулей оперативной памяти и дисковых накопителей.

Десятки мэйнфреймов могут работать совместно под управлением одной операционной системы над выполнением единой задачи.



Суперкомпьютер CRAY —1

Суперкомпьютеры — это очень мощные компьютеры с производительностью свыше 100 мегафлопов (1 мегафлоп — миллион операций с плавающей точкой в секунду). Они называются **сверхбыстродействующими**. Эти машины представляют собой **многопроцессорные** и (или) **многомашинные** комплексы, работающие на общую память и общее поле внешних устройств. Различают суперкомпьютеры **среднего класса, класса выше среднего и переднего края** (high end).

Архитектура суперкомпьютеров основана на идеях **параллелизма** и **конвейеризации вычислений**.

В этих машинах параллельно, то есть одновременно, выполняется множество похожих операций (это называется **мультипроцессорной обработкой**). Таким образом,

сверхвысокое быстродействие обеспечивается **не для всех задач, а только для задач, поддающихся распараллеливанию.**

Что такое **конвейерная обработка**? Приведем сравнение — на каждом рабочем месте конвейера выполняется один шаг производственного процесса, а на всех рабочих местах в одно и то же время обрабатываются различные изделия на всевозможных стадиях. По такому принципу устроено арифметико-логическое устройство суперкомпьютера.

Отличительной особенностью суперкомпьютеров являются **векторные процессоры**, оснащенные аппаратурой для параллельного выполнения операций с многомерными цифровыми объектами — векторами и матрицами. В них встроены **векторные регистры** и **параллельный конвейерный механизм обработки**. Если на обычном процессоре программист выполняет операции над каждым компонентом вектора по очереди, то на векторном — выдаёт сразу векторные команды.

Векторная аппаратура очень дорога, в частности, потому, что требуется много сверхбыстродействующей памяти под векторные регистры.

Наиболее распространённые суперкомпьютеры — **массово-параллельные** компьютерные системы. Они имеют **десятки тысяч процессоров**, взаимодействующих через сложную, иерархически организованную систему памяти.

В качестве примера рассмотрим характеристики многоцелевого массово-параллельного суперкомпьютера среднего класса **Intel Pentium Pro 200**. Этот компьютер содержит 9200 процессоров Pentium Pro на 200 Мгц, в сумме (теоретически) обеспечивающих производительность **1,34 Терафлоп** (1 Терафлоп равен 10^{12} операций с плавающей точкой в секунду), имеет 537 Гбайт памяти и диски ёмкостью 2,25 Терабайт. Система весит 44 тонны (кондиционеры для неё — целых 300 тонн) и потребляет мощность 850 кВт.

Супер-компьютеры используются для решения сложных и больших научных задач (метеорология, гидродинамика и т. п.), в управлении, разведке, в качестве централизованных хранилищ информации и т.д.

Элементная база — микросхемы сверхвысокой степени интеграции.

3.11. Какие существуют типы портативных компьютеров?

Портативные компьютеры обычно нужны руководителям предприятий, менеджерам, учёным, журналистам, которым приходится работать вне офиса — дома, на презентациях или во время командировок.

Основные разновидности портативных компьютеров:



Laptop

Laptop (наколенник, от **lap** — колено и **top** — поверх). По размерам близок к обычному портфелю. По основным характеристикам (быстродействие, память) примерно соответствует настольным ПК. Сейчас компьютеры этого типа уступают место ещё меньшим.



Notebook

Notebook (блокнот, записная книжка). По размерам он ближе к книге крупного формата. Имеет вес около 3 кг. Помещается в портфель-дипломат. Для связи с офисом его обычно комплектуют модемом. Ноутбуки зачастую снабжают приводами CD—ROM.

Многие современные ноутбуки включают взаимозаменяемые блоки со стандартными разъёмами. Такие модули предназначены для очень разных функций. В одно и то же гнездо можно по мере надобности вставлять привод компакт-дисков, накопитель на магнитных дисках, запасную батарею или съёмный винчестер. Ноутбук устойчив к сбоям в энергопитании. Даже если он получает энергию от обычной электросети, в случае какого-либо сбоя он мгновенно переходит на питание от аккумуляторов.



Palmtop



Персональный
цифровой помощник

Palmtop (наладонник) — самые маленькие современные персональные компьютеры. Умещаются на ладони. Магнитные диски в них заменяет энергонезависимая электронная память. Нет и накопителей на дисках — обмен информацией с обычными компьютерами идет линиям связи. Если Palmtop дополнить набором деловых программ, записанных в его постоянную память, получится **персональный цифровой помощник** (Personal Digital Assistant).



Карманный компьютер
iPAQ 3150

Возможности портативных компьютеров постоянно расширяются. Например, современный карманный компьютер **iPAQ 3150** располагает всем необходимым для: ведения списка задач, хранения записок, включая аудиофайлы, работы с календарем, чтения электронной почты, синхронизации с РС, мобильным телефоном. Помимо этого iPAQ позволяет: проигрывать видео и звуковые ролики, бродить по Интернету, просматривать и редактировать документы и электронные таблицы, хранить файлы, искать в них слова, просматривать картинки вести домашнюю бухгалтерию, играть в игры, читать электронные книги с помощью Microsoft Reader, полноценно работать с программным обеспечением.

3.12. Вопросы для самоконтроля

- 3.1. По каким признакам можно разделять компьютеры на классы и виды?
- 3.2. Как эволюционировала элементная база компьютеров от поколения к поколению?
- 3.3. В какой последовательности возникали известные Вам языки программирования?
- 3.4. Когда микрокомпьютеры стали доступны для широкого домашнего применения?
- 3.5. Можете ли Вы связать понятия "яблоко", "гараж" и "компьютер"?
- 3.6. На основе каких технических элементов создавались компьютеры первого поколения?
- 3.7. Какую основную проблему перед разработчиками и пользователями выдвинул опыт эксплуатации компьютеров первого поколения?
- 3.8. Какая элементная база характерна для второго поколения компьютеров?
- 3.9. Какую функцию выполняет операционная система в процессе работы компьютера?
- 3.10. На какой элементной базе конструируются машины третьего поколения?
- 3.11. Из каких основных этапов состоит процесс изготовления микросхем?

3.12. Для каких поколений компьютеров характерно широкое использование интегральных схем?

3.13. Какое быстродействие характерно для машин четвертого поколения?

3.14. Что подразумевают под "интеллектуальностью" компьютеров?

3.15. Какую задачу должен решать "интеллектуальный интерфейс" в машинах пятого поколения?

3.16. Какими особенностями должны обладать промышленные компьютеры?

3.17. Что такое операторский компьютерный интерфейс?

3.18. По каким основным признакам можно отличить мэйнфреймы от других современных компьютеров?

3.19. На какое количество пользователей рассчитаны мэйнфреймы?

3.20. Какие идеи лежат в основе архитектуры суперкомпьютеров?

3.21. На каких типах задач максимально реализуются возможности суперкомпьютеров?

3.22. Какие свойства и конструктивные особенности отличают векторные процессоры?

3.23. Назовите основные характеристики какого-либо суперкомпьютера.

3.24. Что означают в переводе на русский язык названия Laptop, Notebook, Palmtop?

3.25. Как в Palmtop компенсируется отсутствие накопителей на дисках?



Глава 4. Арифметические основы компьютеров

4.1. Что такое система счисления?

Система счисления — это совокупность приемов и правил, по которым числа записываются и читаются.

Существуют позиционные и непозиционные системы счисления.

В непозиционных системах счисления вес цифры (т. е. тот вклад, который она вносит в значение числа) не зависит от ее позиции в записи числа. Так, в римской системе счисления в числе XXXII (тридцать два) вес цифры X в любой позиции равен просто десяти.

В позиционных системах счисления вес каждой цифры изменяется в зависимости от ее положения (позиции) в последовательности цифр, изображающих число. Например, в числе 757,7 первая семерка означает 7 сотен, вторая — 7 единиц, а третья — 7 десятых долей единицы.

Сама же запись числа 757,7 означает сокращенную запись выражения

$$700 + 50 + 7 + 0,7 = 7 \cdot 10^2 + 5 \cdot 10^1 + 7 \cdot 10^0 + 7 \cdot 10^{-1} = 757,7.$$

Любая позиционная система счисления характеризуется своим **основанием**.

Основание позиционной системы счисления — количество различных цифр, используемых для изображения чисел в данной системе счисления.

За основание системы можно принять любое натуральное число — два, три, четыре и т.д. Следовательно, **возможно бесчисленное множество позиционных систем**: двоичная, троичная, четверичная и т.д. Запись чисел в каждой из систем счисления с основанием q означает сокращенную запись выражения

$$a_{n-1} q^{n-1} + a_{n-2} q^{n-2} + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \dots + a_{-m} q^{-m},$$

где a_i — цифры системы счисления; n и m — число целых и дробных разрядов, соответственно.

Например:

Разряды	3 2 1 0 -1
Число	1 0 1 1, $1_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1}$;

Разряды	2 1 0 -1 -2
Число	2 7 6, $5_8 = 2 \cdot 8^2 + 7 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} + 2 \cdot 8^{-2}$.

4.2. Как порождаются целые числа в позиционных системах счисления?

В каждой системе счисления цифры упорядочены в соответствии с их значениями: 1 больше 0, 2 больше 1 и т.д.

Продвижением цифры называют замену её следующей по величине.

Продвинуть цифру 1 значит заменить её на 2, продвинуть цифру 2 значит заменить её на 3 и т.д. **Продвижение старшей цифры** (например, цифры 9 в десятичной системе) означает замену её на 0. В двоичной системе, использующей только две цифры — 0 и 1, продвижение 0 означает замену его на 1, а продвижение 1 — замену её на 0.

Целые числа в любой системе счисления порождаются с помощью **Правила счета** [44]:

Для образования целого числа, следующего за любым данным целым числом, нужно продвинуть самую правую цифру числа; если какая-либо цифра после продвижения стала нулем, то нужно продвинуть цифру, стоящую слева от неё.

Применяя это правило, запишем первые десять целых чисел

- в двоичной системе: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001;
- в троичной системе: 0, 1, 2, 10, 11, 12, 20, 21, 22, 100;
- в пятеричной системе: 0, 1, 2, 3, 4, 10, 11, 12, 13, 14;
- в восьмеричной системе: 0, 1, 2, 3, 4, 5, 6, 7, 10, 11.

4.3. Какие системы счисления используют специалисты для общения с компьютером?

Кроме десятичной широко используются системы с основанием, являющимся целой степенью числа 2, а именно:

- **двоичная** (используются цифры 0, 1);
- **восьмеричная** (используются цифры 0, 1, ..., 7);
- **шестнадцатеричная** (для первых целых чисел от нуля до девяти используются цифры 0, 1, ..., 9, а для следующих чисел — от десяти до пятнадцати — в качестве цифр используются символы A, B, C, D, E, F).

Полезно запомнить запись в этих системах счисления первых двух десятков целых чисел:

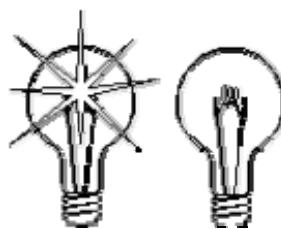
10-я	2-я	8-я	16-я
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9

10-я	2-я	8-я	16-я
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13

Из всех систем счисления **особенно проста** и поэтому **интересна для технической реализации в компьютерах двоичная система счисления**.

4.4. Почему люди пользуются десятичной системой, а компьютеры — двоичной?

Люди предпочитают десятичную систему, вероятно, потому, что с древних времен считали по пальцам, а пальцев у людей по десять на руках и ногах. Не всегда и не везде люди пользуются десятичной системой счисления. В Китае, например, долгое время пользовались пятеричной системой счисления.



А компьютеры используют двоичную систему потому, что она имеет ряд преимуществ перед другими системами:

- для ее реализации нужны **технические устройства с двумя устойчивыми состояниями** (есть ток — нет тока, намагничен — не намагничен и т.п.), а не, например, с десятью, — как в десятичной;
- представление информации посредством только двух состояний **надежно и помехоустойчиво**;

- возможно **применение аппарата булевой алгебры** для выполнения логических преобразований информации;
- двоичная арифметика намного проще десятичной.

Недостаток двоичной системы — **быстрый рост числа разрядов**, необходимых для записи чисел.

4.5. Почему в компьютерах используются также восьмеричная и шестнадцатеричная системы счисления?

Двоичная система, удобная для компьютеров, для человека неудобна из-за ее громоздкости и непривычной записи.

Перевод чисел из десятичной системы в двоичную и наоборот выполняет машина. Однако, чтобы профессионально использовать компьютер, следует научиться понимать слово машины. Для этого и разработаны восьмеричная и шестнадцатеричная системы.

Числа в этих системах читаются почти так же легко, как десятичные, требуют соответственно в три (восьмеричная) и в четыре (шестнадцатеричная) раза меньше разрядов, чем в двоичной системе (ведь числа 8 и 16 — соответственно, третья и четвертая степени числа 2).

Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему очень прост: достаточно каждую цифру заменить эквивалентной ей двоичной триадой (тройкой цифр) или тетрадой (четверкой цифр).

Например:

$$537,1_8 = 101\ 011\ 111,001_2 ; 1A3, F_{16} = 1\ 1010\ 0011, 1111_2$$

↓	↓	↓	↓	↓	↓	↓	↓
5	3	7	1	1	A	3	F

Чтобы перевести число из двоичной системы в восьмеричную или шестнадцатеричную, его нужно разбить влево и вправо от запятой на триады (для восьмеричной) или тетрады (для шестнадцатеричной) и каждую такую группу заменить соответствующей восьмеричной (шестнадцатеричной) цифрой.

Например,

$$10101001,10111_2 = \underset{\downarrow}{10} \ \underset{\downarrow}{101} \ \underset{\downarrow}{001}, \ \underset{\downarrow}{101} \ \underset{\downarrow}{110}_2 = 251,56_8$$

$$10101001,10111_2 = \underset{\downarrow}{1010} \ \underset{\downarrow}{1001}, \ \underset{\downarrow}{1011} \ \underset{\downarrow}{1000}_2 = A9,B8_{16}$$

4.6. Как перевести целое число из десятичной системы в любую другую позиционную систему счисления?

Для перевода целого десятичного числа N в систему счисления с основанием q необходимо N разделить с остатком ("нацело") на q , записанное в той же десятичной системе. Затем неполное частное, полученное от такого деления, нужно снова разделить с остатком на q , и т.д., пока последнее полученное неполное частное не станет равным нулю. Представлением числа N в новой системе счисления будет последовательность остатков деления, изображенных одной q -ичной цифрой и записанных в порядке, обратном порядку их получения.

Пример: Переведем число 75 из десятичной системы в двоичную, восьмеричную и шестнадцатеричную:

В двоичную	В восьмеричную	В шестнадцатеричную
$\begin{array}{r} 75 \mid 2 \\ 1 \quad 37 \mid 2 \\ \quad 1 \quad 18 \mid 2 \\ \quad \quad 0 \quad 9 \mid 2 \\ \quad \quad \quad 1 \quad 4 \mid 2 \\ \quad \quad \quad \quad 0 \quad 2 \mid 2 \\ \quad \quad \quad \quad \quad 0 \quad 1 \mid 2 \\ \quad \quad \quad \quad \quad \quad 0 \quad 1 \mid 2 \\ \quad \quad \quad \quad \quad \quad \quad 0 \end{array}$	$\begin{array}{r} 75 \mid 8 \\ 3 \quad 9 \mid 8 \\ \quad 1 \quad 1 \mid 8 \\ \quad \quad 1 \end{array}$	$\begin{array}{r} 75 \mid 16 \\ (B_{16}) \quad 11 \mid 16 \\ \quad \quad 4 \mid 16 \\ \quad \quad \quad 4 \end{array}$
<p>Напоминание: первый остаток 11₁₀ в этом примере записывается шестнадцатеричной цифрой B₁₆.</p>		

Ответ: $75_{10} = 1 \ 001 \ 011_2 = 113_8 = 4B_{16}$.

4.7. Как перевести правильную десятичную дробь в любую другую позиционную систему счисления?

Для перевода правильной десятичной дроби F в систему счисления с основанием q необходимо F умножить на q , записанное в той же десятичной системе, затем дробную часть полученного произведения снова умножить на q , и т.д., до тех пор, пока дробная часть очередного произведения не станет равной нулю, либо не будет

достигнута требуемая точность изображения числа F в q -ичной системе. Представлением дробной части числа F в новой системе счисления будет последовательность целых частей полученных произведений, записанных в порядке их получения и изображенных одной q -ичной цифрой. Если требуемая точность перевода числа F составляет k знаков после запятой, то предельная абсолютная погрешность при этом равняется $q^{-(k+1)} / 2$.

Пример. Переведем число 0,36 из десятичной системы в двоичную, восьмеричную и шестнадцатеричную:

0,	36	×	2
0	72	×	2
1	44	×	2
0	88	×	2
1	76	×	2
1	52		

Ответ: $0,36_{10} = 0,01011_2$
с предельной абсолютной погрешностью $(2^{-6})/2 = 2^{-7}$.

0,	36	×	8
2	88	×	8
7	04	×	8
0	32		

Ответ: $0,36_{10} = 0,270_8$ с предельной абсолютной погрешностью $(8^{-4})/2 = 2^{-13}$.

0,	36	×	16
5	76	×	16
(C ₁₆)	12	×	16
	8		

Ответ: $0,36_{10} = 0,5C_{16}$ с предельной абсолютной погрешностью $(16^{-3})/2 = 2^{-13}$.

Для чисел, имеющих как целую, так и дробную части, перевод из десятичной системы счисления в другую осуществляется отдельно для целой и дробной частей по правилам, указанным выше.

4.8. Как перевести число из двоичной (восьмеричной, шестнадцатеричной) системы в десятичную?

Перевод в десятичную систему числа x , записанного в q -ичной системе счисления ($q = 2, 8$ или 16) в виде $x_q = (a_n a_{n-1} \dots a_0, a_{-1} a_{-2} \dots a_{-m})_q$ сводится к вычислению значения многочлена

$$x_{10} = a_n q^n + a_{n-1} q^{n-1} + \dots + a_0 q^0 + a_{-1} q^{-1} + a_{-2} q^{-2} + \dots + a_{-m} q^{-m}$$

средствами десятичной арифметики.

Примеры:

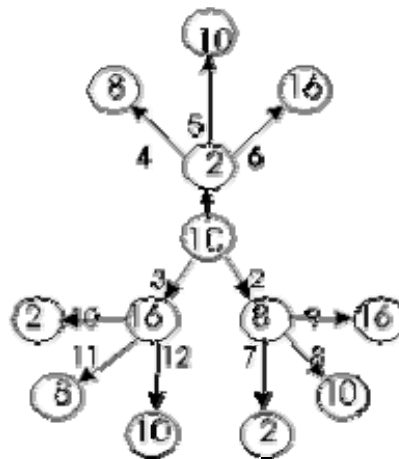
Разряды 3 2 1 0 -1
 Число 1 0 1 1, 1₂ = 1*2³+1*2¹+1*2⁰+1*2⁻¹ = 11,5₁₀

Разряды 2 1 0 -1
 Число 2 7 6, 5₈ = 2*8²+7*8¹+6*8⁰+5*8⁻¹ = 190,625₁₀

Разряды 2 1 0
 Число 1 F 3₁₆ = 1*16²+15*16¹+3*16⁰ = 499₁₀

4.9. Сводная таблица переводов целых чисел из одной системы счисления в другую

Рассмотрим только те системы счисления, которые применяются в компьютерах — десятичную, двоичную, восьмеричную и шестнадцатеричную. Для определенности возьмем произвольное десятичное число, например 46, и для него выполним все возможные последовательные переводы из одной системы счисления в другую. Порядок переводов определим в соответствии с рисунком:



На этом рисунке использованы следующие обозначения:

- в кружках записаны основания систем счисления;
- стрелки указывают направление перевода;
- номер рядом со стрелкой означает порядковый номер соответствующего примера в сводной таблице 4.1.

Например: $\overset{6}{\textcircled{2}} \rightarrow \textcircled{16}$ означает перевод из двоичной системы в шестнадцатеричную, имеющий в таблице порядковый номер 6.

Сводная таблица переводов целых чисел

Таблица 4.1.

№ п/п	Перевод	№ п/п	Перевод
1	$10 \rightarrow 2$ $\begin{array}{r} 46 \overline{) 2} \\ \underline{0} \\ 23 \overline{) 2} \\ \underline{1} \\ 11 \overline{) 2} \\ \underline{1} \\ 5 \overline{) 2} \\ \underline{1} \\ 2 \overline{) 2} \\ \underline{0} \\ 2 \overline{) 2} \\ \underline{0} \\ 1 \end{array}$	5	$2 \rightarrow 10$ $\begin{array}{r} 5 \ 4 \ 3 \ 2 \ 1 \ 0 \\ 1 \ 0 \ 1 \ 1 \ 1 \ 0_2 = 2^5 + 2^3 + 2^2 + 2^1 = \\ = 46_{10} \\ \text{Ответ: } 46_{10} \end{array}$
	<p>Ответ: 101110_2</p>	6	$2 \rightarrow 16$ $101110_2 = 10 \ 1110_2 = 2E_{16}$ <p>Ответ: $2E_{16}$</p>
		7	$8 \rightarrow 2$ $56_8 = 101 \ 110_2$ <p>Ответ: 101110_2</p>
2	$10 \rightarrow 8$ $\begin{array}{r} 46 \overline{) 8} \\ \underline{6} \\ 5 \end{array}$	8	$8 \rightarrow 10$ $\begin{array}{r} 1 \ 0 \\ 5 \ 6_8 = 5 \cdot 8^1 + 6 \cdot 8^0 = 40 + 6 = \\ = 46_{10} \\ \text{Ответ: } 46_{10} \end{array}$
	<p>Ответ: 56_8</p>	9	$8 \rightarrow 16$ $5 \ 6_8 = 101 \ 110_2 = 10 \ 1110_2 = 2E_{16}$ <p>Ответ: $2E_{16}$</p>
№ п/п	Перевод	№ п/п	Перевод
3	$10 \rightarrow 16$ $\begin{array}{r} 46 \overline{) 16} \\ \underline{14} \\ 2 \end{array}$	10	$16 \rightarrow 2$ $2E_{16} = 0010 \ 1110_2 = 101110_2$ <p>Ответ: 101110_2</p>
	<p>Ответ: $2E_{16}$</p>	11	$16 \rightarrow 8$ $2E_{16} = 10 \ 1110_2 = 101 \ 110_2 = 56_8$ <p>Ответ: 56_8</p>
4	$2 \rightarrow 8$ $101110_2 = \underbrace{101}_5 \ \underbrace{110}_6_2 = 56_8$ <p>Ответ: 56_8</p>	12	$16 \rightarrow 10$ $\begin{array}{r} 1 \ 0 \\ 2 \ E_{16} = 2 \cdot 16^1 + E \cdot 16^0 = \\ = 32 + 14 = 46_{10} \\ \text{Ответ: } 46_{10} \end{array}$

4.10. Как производятся арифметические операции в позиционных системах счисления?

Рассмотрим основные арифметические операции: **сложение, вычитание, умножение и деление**. Правила выполнения этих операций в десятичной системе хорошо известны — это сложение, вычитание, умножение столбиком и деление углом. Эти правила применимы и ко всем другим позиционным системам счисления. Только таблицами сложения и умножения надо пользоваться особыми для каждой системы.

Сложение

Таблицы сложения легко составить, используя Правило Счета.

Сложение в двоичной системе

+	0	1
0	0	1
1	1	10

Сложение в восьмеричной системе

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Сложение в шестнадцатеричной системе

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

При сложении цифры суммируются по разрядам, и если при этом возникает избыток, то он переносится влево.

Пример 1. Сложим числа 15 и 6 в различных системах счисления.

<p>Десятичная: $15_{10} + 6_{10}$</p> $\begin{array}{r} 1 \\ + 15 \\ + 6 \\ \hline 21 \\ \hline \end{array}$ <p style="margin-left: 20px;">$5+6=11=10+1$ $1+1=2$</p>	<p>Двоичная: $1111_2 + 110_2$</p> $\begin{array}{r} 111 \\ + 1111 \\ + 0110 \\ \hline 10101 \\ \hline \end{array}$ <p style="margin-left: 20px;">$1+0=1$ $1+1=2=2+0$ $1+1+1=3=2+1$ $1+1=2=2+0$</p>	<p>Восьмеричная: $17_8 + 6_8$</p> $\begin{array}{r} 1 \\ + 17 \\ + 6 \\ \hline 25 \\ \hline \end{array}$ <p style="margin-left: 20px;">$7+6=13=8+5$ $1+1=2$</p>
---	---	--

Шестнадцатеричная: $F_{16} + 6_{16}$

$$\begin{array}{r} 1 \\ + F \\ + 6 \\ \hline 15 \\ \hline \end{array}$$

$15+6=21=16+5$

Ответ: $15+6 = 21_{10} = 10101_2 = 25_8 = 15_{16}$.

Проверка. Преобразуем полученные

суммы к десятичному виду:

$$10101_2 = 2^4 + 2^2 + 2^0 = 16+4+1=21,$$

$$25_8 = 2 \cdot 8^1 + 5 \cdot 8^0 = 16 + 5 = 21,$$

$$15_{16} = 1 \cdot 16^1 + 5 \cdot 16^0 = 16+5 = 21.$$

Пример 2. Сложим числа 15, 7 и 3.

Десятичная: $15_{10} + 7_{10} + 3_{10}$ Двоичная: $1111_2 + 111_2 + 11_2$ Восьмеричная: $17_8 + 7_8 + 3_8$

<p>Десятичная: $15_{10} + 7_{10} + 3_{10}$</p> $\begin{array}{r} 1 \\ + 15 \\ + 7 \\ + 3 \\ \hline 25 \\ \hline \end{array}$ <p style="margin-left: 20px;">$5+7+3=15=10+5$ $1+1=2$</p>	<p>Двоичная: $1111_2 + 111_2 + 11_2$</p> $\begin{array}{r} 11+111 \\ + 1111 \\ + 111 \\ + 11 \\ \hline 11001 \\ \hline \end{array}$ <p style="margin-left: 20px;">$1+1+1=3=2+1$ $1+1+1+1=4=2 \cdot 2+0$ $1+1=2=2+0$ $1+1+1=3=2+1$</p>	<p>Восьмеричная: $17_8 + 7_8 + 3_8$</p> $\begin{array}{r} 2 \\ + 17 \\ + 7 \\ + 3 \\ \hline 31 \\ \hline \end{array}$ <p style="margin-left: 20px;">$7+7+3=17=2 \cdot 8+1$ $2+1=3$</p>
---	--	---

Шестнадцатеричная: $F_{16} + 7_{16} + 3_{16}$

$$\begin{array}{r} + F \\ + 7 \\ + 3 \\ \hline 19 \\ \hline \end{array}$$

$15+7+3=25=16+9$

Ответ: $5+7+3 = 25_{10} = 11001_2 = 31_8 = 19_{16}$.

Проверка:

$$11001_2 = 2^4 + 2^3 + 2^0 = 16+8+1=25,$$

$$31_8 = 3 \cdot 8^1 + 1 \cdot 8^0 = 24 + 1 = 25,$$

$$19_{16} = 1 \cdot 16^1 + 9 \cdot 16^0 = 16+9 = 25.$$

Пример 3. Сложим числа 141,5 и 59,75.

Десятичная: $141,5_{10} + 59,75_{10}$

$$\begin{array}{r}
 111 \\
 + 141,50 \\
 \quad 59,75 \\
 \hline
 201,25 \\
 \hline
 \begin{array}{l}
 \boxed{0+5=5} \\
 \boxed{5+7=12=10+2} \\
 \boxed{1+9+1=11=10+1} \\
 \boxed{4+5+1=10=10+0} \\
 \boxed{1+1=2}
 \end{array}
 \end{array}$$

Двоичная: $10001101,1_2 + 111011,11_2$

$$\begin{array}{r}
 1111111 \\
 + 10001101,1 \\
 \quad 111011,11 \\
 \hline
 11001001,01 \\
 \hline
 \begin{array}{l}
 \boxed{1+0=1} \\
 \boxed{1+1=2=2+0} \\
 \boxed{1+1=2=2+0} \\
 \boxed{1+1+1=3=2+1} \\
 \boxed{1+1=2=2+0} \\
 \boxed{1+1=2=2+0}
 \end{array}
 \end{array}$$

Восьмеричная: $215,4_8 + 73,6_8$

$$\begin{array}{r}
 111 \\
 + 215,4 \\
 \quad 73,6 \\
 \hline
 311,2 \\
 \hline
 \begin{array}{l}
 \boxed{4+6=10=8+2} \\
 \boxed{5+3+1=9=8+1} \\
 \boxed{1+7+1=9=8+1} \\
 \boxed{2+1=3}
 \end{array}
 \end{array}$$

Шестнадцатеричная: $8D,8_{16} + 3B,C_{16}$

$$\begin{array}{r}
 11 \\
 + 8D,8 \\
 \quad 3B,C \\
 \hline
 C9,4 \\
 \hline
 \begin{array}{l}
 \boxed{8+12=20=16+4} \\
 \boxed{13+11+1=25=16+9} \\
 \boxed{8+3+1=12=C_{16}}
 \end{array}
 \end{array}$$

Ответ: $141,5 + 59,75 = 201,25_{10} = 11001001,01_2 = 311,2_8 = C9,4_{16}$

Проверка. Преобразуем полученные суммы к десятичному виду:
 $11001001,01_2 = 2^7 + 2^6 + 2^3 + 2^0 + 2^{-2} = 201,25$
 $311,2_8 = 3 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0 + 2 \cdot 8^{-1} = 201,25$
 $C9,4_{16} = 12 \cdot 16^1 + 9 \cdot 16^0 + 4 \cdot 16^{-1} = 201,25$

Вычитание

Пример 4. Вычтем единицу из чисел 10_2 , 10_8 и 10_{16}

Двоичная: $10_2 - 1_2$

$$\begin{array}{r}
 1 \\
 - 10 \\
 \quad 1 \\
 \hline
 1 \\
 \hline
 \boxed{2-1=1}
 \end{array}$$

Восьмеричная: $10_8 - 1_8$

$$\begin{array}{r}
 1 \\
 - 10 \\
 \quad 1 \\
 \hline
 7 \\
 \hline
 \boxed{8-1=7}
 \end{array}$$

Шестнадцатеричная: $10_{16} - 1_{16}$

$$\begin{array}{r}
 1 \quad \text{Заемы} \\
 - 10 \\
 \quad 1 \\
 \hline
 F \\
 \hline
 \boxed{16-1=15=F_{16}}
 \end{array}$$

Пример 5. Вычтем единицу из чисел 100_2 , 100_8 и 100_{16} .

Двоичная: $100_2 - 1_2$

$$\begin{array}{r}
 1 \\
 - 100 \\
 \quad 1 \\
 \hline
 11 \\
 \hline
 \boxed{2-1=1} \\
 \boxed{1-0=1}
 \end{array}$$

Восьмеричная: $100_8 - 1_8$

$$\begin{array}{r}
 1 \\
 - 100 \\
 \quad 1 \\
 \hline
 77 \\
 \hline
 \boxed{8-1=7} \\
 \boxed{7-0=7}
 \end{array}$$

Шестнадцатеричная: $100_{16} - 1_{16}$

$$\begin{array}{r}
 1 \quad \text{Заемы} \\
 - 100 \\
 \quad 1 \\
 \hline
 FF \\
 \hline
 \boxed{16-1=15=F_{16}} \\
 \boxed{1+1=2}
 \end{array}$$

Ввиду чрезвычайной простоты таблицы умножения в двоичной системе, умножение сводится лишь к сдвигам множимого и сложениям.

Пример 7. Перемножим числа 5 и 6.

Десятичная: $5_{10} \cdot 6_{10}$ Двоичная: $101_2 \cdot 110_2$ Восьмеричная: $5_8 \cdot 6_8$

$$\begin{array}{r} \times 5 \\ 6 \\ \hline 30 \end{array}$$

$$\begin{array}{r} \times 101 \\ 110 \\ \hline 101 \\ 101 \\ \hline 11110 \end{array}$$

$$\begin{array}{r} \times 5 \\ 6 \\ \hline 36 \end{array}$$

Ответ: $5 \cdot 6 = 30_{10} = 11110_2 = 36_8$.

Проверка. Преобразуем полученные произведения к десятичному виду:

$$11110_2 = 2^4 + 2^3 + 2^2 + 2^1 = 30;$$

$$36_8 = 3 \cdot 8^1 + 6 \cdot 8^0 = 30.$$

Пример 8. Перемножим числа 115 и 51.

Десятичная: $115_{10} \cdot 51_{10}$ Двоичная: $1110011_2 \cdot 110011_2$ Восьмеричная: $163_8 \cdot 63_8$

$$\begin{array}{r} \times 115 \\ 51 \\ \hline 115 \\ 575 \\ \hline 5865 \end{array}$$

$$\begin{array}{r} \times 1110011 \\ 110011 \\ \hline 1110011 \\ 1110011 \\ \hline 1110011 \\ 1110011 \\ \hline 1011011101001 \end{array}$$

$$\begin{array}{r} \times 163 \\ 63 \\ \hline 531 \\ 1262 \\ \hline 13351 \end{array}$$

Ответ: $115 \cdot 51 = 5865_{10} = 1011011101001_2 = 13351_8$.

Проверка. Преобразуем полученные произведения к десятичному виду:

$$1011011101001_2 = 2^{12} + 2^{10} + 2^9 + 2^7 + 2^6 + 2^5 + 2^3 + 2^0 = 5865;$$

$$13351_8 = 1 \cdot 8^4 + 3 \cdot 8^3 + 3 \cdot 8^2 + 5 \cdot 8^1 + 1 \cdot 8^0 = 5865.$$

Деление

Деление в любой позиционной системе счисления производится по тем же правилам, как и деление углом в десятичной системе. В двоичной системе деление выполняется особенно просто, ведь очередная цифра частного может быть только нулем или единицей.

Пример 9. Разделим число 30 на число 6.

Десятичная: $30_{10} : 6_{10}$ Двоичная: $11110_2 : 110_2$ Восьмеричная: $36_8 : 6_8$

$$\begin{array}{r} - 30 \overline{) 6} \\ 30 \overline{) 5} \\ \hline 0 \end{array}$$

$$\begin{array}{r} - 11110 \overline{) 110} \\ 110 \overline{) 101} \\ \hline 110 \\ 110 \\ \hline 0 \end{array}$$

$$\begin{array}{r} - 36 \overline{) 6} \\ 36 \overline{) 5} \\ \hline 0 \end{array}$$

Ответ: $30 : 6 = 5_{10} = 101_2 = 5_8$.

Пример 10. Разделим число 5865 на число 115.

Десятичная: $5865_{10} : 115_{10}$ Двоичная: $1011011101001_2 : 1110011_2$

$$\begin{array}{r} \underline{5865} \mid \underline{115} \\ \underline{575} \quad \underline{51} \\ \underline{115} \\ \underline{115} \\ \underline{0} \end{array}$$

$$\begin{array}{r} \underline{1011011101001} \mid \underline{1110011} \\ \underline{1110011} \quad \underline{110011} \\ \underline{1000100} \\ \underline{1110011} \\ \underline{10101100} \\ \underline{1110011} \\ \underline{1110011} \\ \underline{1110011} \\ \underline{0} \end{array}$$

Восьмеричная: $13351_8 : 163_8$

$$\begin{array}{r} \underline{13351} \mid \underline{163} \\ \underline{1262} \quad \underline{63} \\ \underline{531} \\ \underline{531} \\ \underline{0} \end{array}$$

Ответ: $5865 : 115 = 51_{10} = 110011_2 = 63_8$.

Проверка. Преобразуем полученные частные к десятичному виду:
 $110011_2 = 2^5 + 2^4 + 2^1 + 2^0 = 51$; $63_8 = 6 \cdot 8^1 + 3 \cdot 8^0 = 51$.

Пример 11. Разделим число 35 на число 14.

Десятичная: $35_{10} : 14_{10}$

Двоичная: $100011_2 : 1110_2$

$$\begin{array}{r} \underline{35} \mid \underline{14} \\ \underline{28} \quad \underline{2,5} \\ \underline{70} \\ \underline{70} \\ \underline{0} \end{array}$$

$$\begin{array}{r} \underline{100011} \mid \underline{1110} \\ \underline{1110} \quad \underline{10,1} \\ \underline{1110} \\ \underline{1110} \\ \underline{0} \end{array}$$

Восьмеричная: $43_8 : 16_8$

$$\begin{array}{r} \underline{43} \mid \underline{16} \\ \underline{34} \quad \underline{2,4} \\ \underline{70} \\ \underline{70} \\ \underline{0} \end{array}$$

Ответ: $35 : 14 = 2,5_{10} = 10,1_2 = 2,4_8$.

Проверка. Преобразуем полученные частные к десятичному виду:
 $10,1_2 = 2^1 + 2^{-1} = 2,5$;
 $2,4_8 = 2 \cdot 8^0 + 4 \cdot 8^{-1} = 2,5$.

4.11. Как представляются в компьютере целые числа?

Целые числа могут представляться в компьютере со знаком или без знака.

Целые числа без знака

Обычно занимают в памяти компьютера один или два байта. В однобайтовом формате принимают значения от 00000000_2 до 11111111_2 . В двухбайтовом формате

- от 00000000 00000000₂ до 11111111 11111111₂.

Диапазоны значений целых чисел без знака

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	0 ... 2 ⁸ -1	0 ... 255
2	0 ... 2 ¹⁶ -1	0 ... 65535

Примеры:

а) число 72₁₀ = 1001000₂ в **однобайтовом** формате:

Номера разрядов: 7 6 5 4 3 2 1 0
 Биты числа: 0 1 0 0 1 0 0 0

б) это же число в **двубайтовом** формате:

Номера разрядов: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
 Биты числа: 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0

в) число 65535 в **двубайтовом** формате:

Номера разрядов: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
 Биты числа: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Целые числа со знаком

Обычно занимают в памяти компьютера один, два или четыре байта, при этом самый левый (старший) разряд содержит информацию о знаке числа.

Диапазоны значений целых чисел со знаком

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	-2 ⁷ ... 2 ⁷ -1	-128 ... 127
2	-2 ¹⁵ ... 2 ¹⁵ -1	-32768 ... 32767
4	-2 ³¹ ... 2 ³¹ -1	-2147483648 ... 2147483647

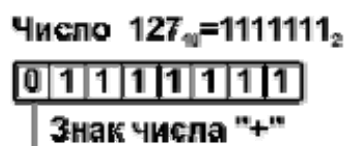
Рассмотрим особенности записи целых чисел со знаком на примере **однобайтового формата**, при котором для знака отводится один разряд, а для цифр абсолютной величины - семь разрядов.

В компьютерной технике применяются три формы записи (кодирования) целых чисел со знаком:

прямой код, **обратный код**, **дополнительный код**.

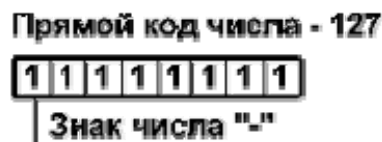
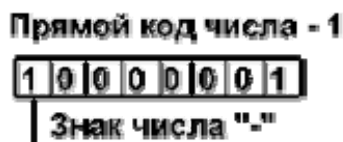
Последние две формы применяются особенно широко, так как позволяют упростить конструкцию арифметико-логического устройства компьютера путем замены разнообразных арифметических операций операцией сложения.

Положительные числа в прямом, обратном и дополнительном кодах изображаются одинаково - двоичными кодами с цифрой 0 в знаковом разряде. Например:

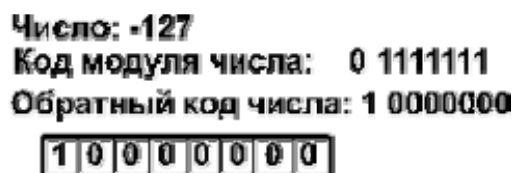
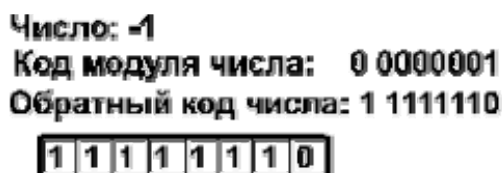


Отрицательные числа в прямом, обратном и дополнительном кодах имеют разное изображение.

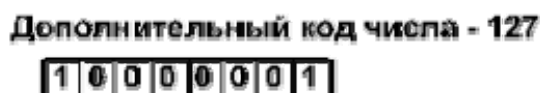
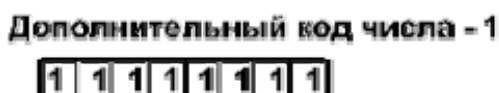
1. **Прямой код**. В знаковый разряд помещается цифра 1, а в разряды цифровой части числа — двоичный код его абсолютной величины. Например:



2. **Обратный код**. Получается инвертированием всех цифр двоичного кода абсолютной величины числа, включая разряд знака: нули заменяются единицами, а единицы — нулями. Например:



3. **Дополнительный код**. Получается образованием обратного кода с последующим прибавлением единицы к его младшему разряду. Например:



Обычно отрицательные десятичные числа при вводе в машину автоматически преобразуются в обратный или дополнительный двоичный код и в таком виде хранятся, перемещаются и участвуют в операциях. При выводе таких чисел из машины происходит обратное преобразование в отрицательные десятичные числа.

4.12. Как компьютер выполняет арифметические действия над целыми числами?

Сложение и вычитание

В большинстве компьютеров операция вычитания не используется. Вместо нее производится сложение обратных или дополнительных кодов уменьшаемого и вычитаемого. Это позволяет существенно упростить конструкцию АЛУ.

Сложение обратных кодов. Здесь при сложении чисел А и В имеют место четыре основных и два особых случая:

1. А и В положительные. При суммировании складываются все разряды, включая разряд знака. Так как знаковые разряды положительных слагаемых равны нулю, разряд знака суммы тоже равен нулю. Например:

Десятичная запись

$$\begin{array}{r} + 3 \\ + 7 \\ \hline 10 \end{array}$$

Двоичные коды

$$\begin{array}{r} + 0000011 \\ + 0000111 \\ \hline 0001010 \end{array}$$

Получен правильный результат.

2. А положительное, В отрицательное и по абсолютной величине больше, чем А. Например:

Десятичная запись

$$\begin{array}{r} + 3 \\ + -10 \\ \hline -7 \end{array}$$

Двоичные коды

$$\begin{array}{r} + 0000011 \\ + 11110101 \\ \hline 1111000 \end{array}$$

Обратный код числа -10
Обратный код числа -7

Получен правильный результат в обратном коде. При переводе в прямой код биты цифровой части результата инвертируются: $1000111 = -7_{10}$.

3. А положительное, В отрицательное и по абсолютной величине меньше, чем А. Например:

Десятичная запись

$$\begin{array}{r} + 10 \\ + -3 \\ \hline 7 \end{array}$$

Двоичные коды

$$\begin{array}{r} + 0001010 \\ + 1111100 \\ \hline 0000110 \\ \rightarrow +1 \\ \hline 0000111 \end{array}$$

Обратный код числа -3

Компьютер исправляет полученный первоначально неправильный результат (6 вместо 7) **переносом единицы** из знакового разряда в младший разряд суммы.

4. А и В отрицательные. Например:

Десятичная запись	Двоичные коды
$\begin{array}{r} + -3 \\ + -7 \\ \hline -10 \end{array}$	$\begin{array}{r} + 1\ 111100 \\ + 1\ 111000 \\ \hline 1\ 1110100 \\ \xrightarrow{+1} \\ \hline 1\ 1110101 \end{array}$ <p>Обратный код числа -3 Обратный код числа -7 Обратный код числа -10</p>

Полученный первоначально неправильный результат (обратный код числа -11_{10} вместо обратного кода числа -10_{10}) компьютер исправляет переносом единицы из знакового разряда в младший разряд суммы. При переводе результата в прямой код биты цифровой части числа инвертируются: $1\ 0001010 = -10_{10}$.

При сложении может возникнуть ситуация, когда старшие разряды результата операции не помещаются в отведенной для него области памяти. Такая ситуация называется **переполнением разрядной сетки формата числа**. Для обнаружения переполнения и оповещения о возникшей ошибке в компьютере используются специальные средства. Ниже приведены два возможных случая переполнения.

5. А и В положительные, сумма А+В больше, либо равна 2^{n-1} , где n — количество разрядов формата чисел (для однобайтового формата $n=8$, $2^{n-1} = 2^7 = 128$). Например:

Десятичная запись	Двоичные коды
$\begin{array}{r} + 65 \\ + 97 \\ \hline 162 \end{array}$	$\begin{array}{r} + 0\ 1000001 \\ + 0\ 1100001 \\ \hline 1\ 0100010 \end{array}$ <p>Переполнение</p>

Семи разрядов цифровой части числового формата **недостаточно** для размещения восьмиразрядной суммы ($162_{10} = 10100010_2$), поэтому **старший разряд суммы оказывается в знаковом разряде**. Это вызывает **несовпадение знака суммы и знаков слагаемых**, что является свидетельством переполнения разрядной сетки.

6. А и В отрицательные, сумма абсолютных величин А и В больше, либо равна 2^{n-1} . Например:

Десятичная запись	Двоичные коды
$\begin{array}{r} + -63 \\ + -95 \\ \hline 158 \end{array}$	$\begin{array}{r} + 1\ 1000000 \\ + 1\ 0100000 \\ \hline 0\ 1100000 \\ \xrightarrow{+1} \\ \hline \end{array}$ <p>Обратный код числа -63 Обратный код числа -95 Переполнение</p>

Здесь знак суммы тоже не совпадает со знаками слагаемых, что свидетельствует о переполнении разрядной сетки.

Сложение дополнительных кодов. Здесь также имеют место рассмотренные выше шесть случаев:

1. А и В положительные. Здесь нет отличий от случая 1, рассмотренного для обратного кода.

2. А положительное, В отрицательное и по абсолютной величине больше, чем А. Например:

Десятичная запись	Двоичные коды
$\begin{array}{r} + 3 \\ -10 \\ \hline -7 \end{array}$	$\begin{array}{r} + 0\ 0000011 \\ 1\ 1110110 \\ \hline 1\ 1111001 \end{array}$
	Дополнительный код числа -10 Дополнительный код числа -7

Получен правильный результат в дополнительном коде. При переводе в прямой код биты цифровой части результата инвертируются и к младшему разряду прибавляется единица: $1\ 0000110 + 1 = 1\ 0000111 = -7_{10}$.

3. А положительное, В отрицательное и по абсолютной величине меньше, чем А. Например:

Десятичная запись	Двоичные коды
$\begin{array}{r} + 10 \\ -3 \\ \hline 7 \end{array}$	$\begin{array}{r} + 0\ 0001010 \\ 1\ 1111101 \\ \hline 0\ 0000111 \end{array}$
	Дополнительный код числа -3 ————— перенос отбрасывается

Получен правильный результат. Единицу переноса из знакового разряда компьютер отбрасывает.

4. А и В отрицательные. Например:

Десятичная запись	Двоичные коды
$\begin{array}{r} + -3 \\ -7 \\ \hline -10 \end{array}$	$\begin{array}{r} + 1\ 1111101 \\ 1\ 1111001 \\ \hline 1\ 1110110 \end{array}$
	Дополнительный код числа -3 Дополнительный код числа -7 ————— перенос отбрасывается

Получен правильный результат в дополнительном коде. Единицу переноса из знакового разряда компьютер отбрасывает.

Случаи переполнения для дополнительных кодов рассматриваются по аналогии со случаями 5 и 6 для обратных кодов.

Сравнение рассмотренных форм кодирования целых чисел со знаком показывает:

- на преобразование отрицательного числа в обратный код компьютер затрачивает меньше времени, чем на преобразование в дополнительный код, так как последнее состоит из двух шагов — образования обратного кода и прибавления единицы к его младшему разряду;
- время выполнения сложения для дополнительных кодов чисел меньше, чем для их обратных кодов, потому что в таком сложении нет переноса единицы из знакового разряда в младший разряд результата.

Умножение и деление

Во многих компьютерах **умножение** производится как последовательность сложений и сдвигов. Для этого в АЛУ имеется **регистр**, называемый **накапливающим сумматором**, который до начала выполнения операции **содержит число ноль**. В процессе выполнения операции в нем поочередно размещаются **множимое** и **результаты промежуточных сложений**, а по завершении операции — **окончательный результат**.

Другой регистр АЛУ, участвующий в выполнении этой операции, **вначале содержит множитель**. Затем по мере выполнения сложений содержащееся в нем **число уменьшается, пока не достигнет нулевого значения**.

Для иллюстрации умножим 110011_2 на 101101_2 .

Накапливающий сумматор	Множитель
$\begin{array}{r} 000000000000 \\ + \quad 110011 \\ \hline 110011 \\ + \quad 110011 \\ \hline 11111111 \\ + \quad 110011 \\ \hline 1010010111 \\ + \quad 110011 \\ \hline 100011110111 \end{array}$	$\begin{array}{r} 101101 \\ 101100 \\ 101000 \\ 100000 \\ 000000 \end{array}$ <p>Сдвиг на две позиции влево Сдвиг на одну позицию влево Сдвиг на две позиции влево</p>

Деление для компьютера является трудной операцией. Обычно оно реализуется путем многократного прибавления к делимому дополнительного кода делителя.

4.13. Как представляются в компьютере вещественные числа?

Система вещественных чисел в математических вычислениях предполагается непрерывной и бесконечной, т.е. не имеющей ограничений на диапазон и точность представления чисел. Однако в компьютерах числа хранятся в регистрах и ячейках памяти с ограниченным количеством разрядов. В следствие этого **система вещественных чисел, представимых в машине, является дискретной (прерывной) и конечной**.

При написании вещественных чисел в программах вместо привычной запятой принято ставить точку. Для отображения вещественных чисел, которые могут быть как очень маленькими, так и очень большими, используется форма записи чисел с **порядком основания системы счисления**. Например, десятичное число 1.25 в этой форме можно представить так:

$$1.25 \cdot 10^0 = 0.125 \cdot 10^1 = 0.0125 \cdot 10^2 = \dots$$

или так:

$$12.5 \cdot 10^{-1} = 125.0 \cdot 10^{-2} = 1250.0 \cdot 10^{-3} = \dots$$

Любое число N в системе счисления с основанием q можно записать в виде $N = M \cdot q^p$, где

M — множитель, содержащий все цифры числа (**мантисса**), а p — целое число, называемое **порядком**. Такой способ записи чисел называется **представлением числа с плавающей точкой**.

Если "плавающая" точка расположена в мантиссе перед первой значащей цифрой, то при фиксированном количестве разрядов, отведённых под мантиссу, обеспечивается запись максимального количества значащих цифр числа, то есть максимальная точность представления числа в машине. Из этого следует:

Мантисса должна быть правильной дробью, у которой первая цифра после точки (запятой в обычной записи) отлична от нуля: $0.1_2 \leq |M| < 1$. Если это требование выполнено, то число называется нормализованным

Мантиссу и порядок q -ичного числа принято записывать в системе с основанием q , а само основание — в десятичной системе. Примеры нормализованного представления:

Десятичная система

Двоичная система

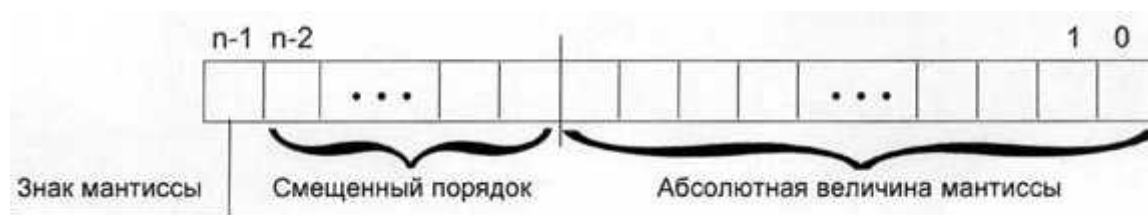
$$753.15 = 0.75315 \cdot 10^3;$$

$$-101.01 = -0.10101 \cdot 2^{11} \text{ (порядок } 11_2 = 3_{10})$$

$$-0.000034 = -0.34 \cdot 10^{-4};$$

$$0.000011 = 0.11 \cdot 2^{-100} \text{ (порядок } -100_2 = -4_{10}).$$

Вещественные числа в компьютерах различных типов записываются по-разному, тем не менее, **все компьютеры поддерживают несколько международных стандартных форматов, различающихся по точности, но имеющих одинаковую структуру следующего вида:**



Здесь порядок n -разрядного нормализованного числа задается в так называемой **смещенной форме**: если для задания порядка выделено k разрядов, то к истинному значению порядка, **представленного в дополнительном коде**, прибавляют смещение, равное $(2^{k-1} - 1)$. Например, порядок, принимающий значения в диапазоне от -128 до $+127$, представляется смещенным порядком, значения которого меняются от 0 до 255 .

Использование смещенной формы позволяет производить операции над порядками, как над беззнаковыми числами, что упрощает операции сравнения, сложения и вычитания порядков, а также упрощает операцию сравнения самих нормализованных чисел.

Чем больше разрядов отводится под запись мантииссы, тем выше точность представления числа. Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего отличного от нуля числа до наибольшего числа, представимого в машине при заданном формате.

Стандартные форматы представления вещественных чисел:

1) **одинарный** — 32-разрядное нормализованное число со знаком, 8-разрядным смещенным порядком и 24-разрядной мантииссой (старший бит мантииссы, всегда равный 1, не хранится в памяти, и размер поля, выделенного для хранения мантииссы, составляет только 23 разряда).

2) **двойной** — 64-разрядное нормализованное число со знаком, 11-разрядным смещенным порядком и 53-разрядной мантииссой (старший бит мантииссы не хранится, размер поля, выделенного для хранения мантииссы, составляет 52 разряда).

3) **расширенный** — 80-разрядное число со знаком, 15-разрядным смещенным порядком и 64-разрядной мантииссой. Позволяет хранить ненормализованные числа.

Следует отметить, что вещественный формат с m -разрядной мантииссой позволяет абсолютно точно представлять m -разрядные целые числа, т. е. **любое двоичное целое число, содержащее не более m разрядов, может быть без искажений преобразовано в вещественный формат.**

4.14. Как компьютер выполняет арифметические действия над нормализованными числами?

К началу выполнения арифметического действия операнды операции помещаются в соответствующие регистры АЛУ.

Сложение и вычитание

При сложении и вычитании сначала производится подготовительная операция, называемая **выравниванием порядков**.

В процессе выравнивания порядков мантиисса числа с меньшим порядком **сдвигается в своем регистре вправо** на количество разрядов, равное разности порядков операндов. После каждого сдвига **порядок увеличивается на единицу**.

В результате выравнивания порядков одноименные разряды чисел оказываются расположенными в соответствующих разрядах обоих регистров, после чего **мантииссы складываются или вычитаются**. В случае необходимости полученный результат нормализуется путем сдвига мантииссы результата влево. После каждого сдвига влево порядок результата уменьшается на единицу.

Пример 1. Сложить двоичные нормализованные числа $0.10111 \cdot 2^{-1}$ и $0.11011 \cdot 2^{10}$. Разность порядков слагаемых здесь равна трем, поэтому перед сложением мантисса первого числа сдвигается на три разряда вправо:

$$\begin{array}{r}
 + \quad 0.00010111 \cdot 2^{10} \\
 \quad 0.11011 \quad \cdot 2^{10} \\
 \hline
 \quad 0.11101111 \cdot 2^{10}
 \end{array}$$

Пример 2. Выполнить вычитание двоичных нормализованных чисел $0.10101 \cdot 2^{10}$ и $0.11101 \cdot 2^1$. Разность порядков уменьшаемого и вычитаемого здесь равна единице, поэтому перед вычитанием мантисса второго числа сдвигается на один разряд вправо:

$$\begin{array}{r}
 - \quad 0.10101 \quad \cdot 2^{10} \\
 \quad 0.011101 \quad \cdot 2^{10} \\
 \hline
 \quad 0.001101 \quad \cdot 2^{10}
 \end{array}$$

Результат получился **не нормализованным**, поэтому его **мантисса сдвигается влево на два разряда** с соответствующим уменьшением порядка на две единицы: $0.1101 \cdot 2^0$.

Умножение

При умножении двух нормализованных чисел их порядки складываются, а мантиссы перемножаются.

Пример 3. Выполнить умножение двоичных нормализованных чисел:

$$(0.11101 \cdot 2^{101}) \cdot (0.1001 \cdot 2^{11}) = (0.11101 \cdot 0.1001) \cdot 2^{(101+11)} = 0.100000101 \cdot 2^{1000}.$$

Деление

При делении двух нормализованных чисел из порядка делимого вычитается порядок делителя, а мантисса делимого делится на мантиссу делителя. Затем в случае необходимости полученный результат нормализуется.

Пример 4. Выполнить деление двоичных нормализованных чисел:

$$0.1111 \cdot 2^{100} : 0.101 \cdot 2^{11} = (0.1111 : 0.101) \cdot 2^{(100-11)} = 1.1 \cdot 2^1 = 0.11 \cdot 2^{10}.$$

Использование представления чисел с плавающей точкой существенно усложняет схему арифметико-логического устройства.

4.15. Упражнения

4.1. Используя Правило Счета, запишите первые 20 целых чисел в десятичной, двоичной, троичной, пятеричной и восьмеричной системах счисления.

4.2. Какие целые числа следуют за числами:

- | | | |
|-----------------|---------------|------------------|
| а) 1_2 ; | е) 1_8 ; | п) F_{16} ; |
| б) 101_2 ; | ж) 7_8 ; | м) $1F_{16}$; |
| в) 111_2 ; | з) 37_8 ; | н) FF_{16} ; |
| г) 1111_2 ; | и) 177_8 ; | о) $9AF9_{16}$; |
| д) 101011_2 ; | к) 7777_8 ; | п) $CDEF_{16}$? |

4.3. Какие целые числа предшествуют числам:

- | | | |
|----------------|---------------|------------------|
| а) 10_2 ; | е) 10_8 ; | л) 10_{16} ; |
| б) 1010_2 ; | ж) 20_8 ; | м) 20_{16} ; |
| в) 1000_2 ; | з) 100_8 ; | н) 100_{16} ; |
| г) 10000_2 ; | и) 110_8 ; | о) $A10_{16}$; |
| д) 10100_2 ; | к) 1000_8 ; | п) 1000_{16} ? |

4.4. Какой цифрой заканчивается четное двоичное число? Какой цифрой заканчивается нечетное двоичное число? Какими цифрами может заканчиваться четное троичное число?

4.5. Какое наибольшее десятичное число можно записать тремя цифрами:

- а) в двоичной системе;
- б) в восьмеричной системе;
- в) в шестнадцатеричной системе?

4.6. В какой системе счисления $21 + 24 = 100$?

Решение. Пусть x — искомое основание системы счисления. Тогда $100_x = 1 \cdot x^2 + 0 \cdot x^1 + 0 \cdot x^0$, $21_x = 2 \cdot x^1 + 1 \cdot x^0$, $24_x = 2 \cdot x^1 + 4 \cdot x^0$. Таким образом, $x^2 = 2x + 2x + 5$ или $x^2 - 4x - 5 = 0$. Положительным корнем этого квадратного уравнения является $x = 5$.

Ответ. Числа записаны в пятеричной системе счисления.

4.7. В какой системе счисления справедливо следующее:

- а) $20 + 25 = 100$;
- б) $22 + 44 = 110$?

4.8. Десятичное число 59 эквивалентно числу 214 в некоторой другой системе счисления. Найдите основание этой системы.

4.9. Переведите числа в десятичную систему, а затем проверьте результаты, выполнив обратные переводы:

- | | | |
|--------------------|-----------------|--------------------|
| а) 1011011_2 ; | е) 517_8 ; | л) $1F_{16}$; |
| б) 10110111_2 ; | ж) 1010_8 ; | м) ABC_{16} ; |
| в) 011100001_2 ; | з) 1234_8 ; | н) 1010_{16} ; |
| г) $0,1000110_2$; | и) $0,34_8$; | о) $0,А4_{16}$; |
| д) $110100,11_2$; | к) $123,41_8$; | п) $1DE,C8_{16}$. |

4.10. Переведите числа из десятичной системы в двоичную, восьмеричную и шестнадцатеричную, а затем проверьте результаты, выполнив обратные переводы:

- а) 125_{10} ; б) 229_{10} ; в) 88_{10} ; г) $37,25_{10}$; д) $206,125_{10}$.

4.11. Переведите числа из двоичной системы в восьмеричную и шестнадцатеричную, а затем проверьте результаты, выполнив обратные переводы:

- | | |
|-----------------------------|---------------------------|
| а) $1001111110111,0111_2$; | г) $1011110011100,11_2$; |
| б) $1110101011,1011101_2$; | д) $10111,1111101111_2$; |
| в) $10111001,101100111_2$; | е) $1100010101,11001_2$. |

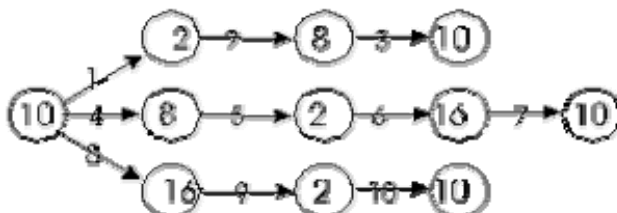
4.12. Переведите в двоичную и восьмеричную системы шестнадцатеричные числа:

- а) $2CE_{16}$; б) $9F40_{16}$; в) $ABCDE_{16}$; г) $1010,101_{16}$; д) $1ABC,9D_{16}$.

4.13. Выпишите целые числа:

- а) от 101101_2 до 110000_2 в двоичной системе;
- б) от 202_3 до 1000_3 в троичной системе;
- в) от 14_8 до 20_8 в восьмеричной системе;
- г) от 28_{16} до 30_{16} в шестнадцатеричной системе.

4.14. Для десятичных чисел 47 и 79 выполните цепочку переводов из одной системы счисления в другую:



4.15. Составьте таблицы сложения однозначных чисел в троичной и пятеричной системах счисления.

4.16. Составьте таблицы умножения однозначных чисел в троичной и пятеричной системах счисления.

4.17. Сложите числа, а затем проверьте результаты, выполнив соответствующие десятичные сложения:

- | | | |
|------------------------------------|-----------------------------|-------------------------------------|
| а) 1011101_2 и 1110111_2 ; | д) 37_8 и 75_8 ; | и) A_{16} и F_{16} ; |
| б) $1011,101_2$ и $101,011_2$; | е) 165_8 и 37_8 ; | к) 19_{16} и C_{16} ; |
| в) 1011_2 , 11_2 и $111,1_2$; | ж) $7,5_8$ и $14,6_8$; | л) A, B_{16} и E, F_{16} ; |
| г) 1011_2 , $11,1_2$ и 111_2 ; | з) 6_8 , 17_8 и 7_8 ; | м) E_{16} , 9_{16} и F_{16} . |

4.18. В каких системах счисления выполнены следующие сложения? Найдите основания каждой системы:

а) $\begin{array}{r} 98 \\ + 89 \\ \hline 121 \end{array}$	б) $\begin{array}{r} 1345 \\ + 2178 \\ \hline 3523 \end{array}$	в) $\begin{array}{r} 10101 \\ + 1111 \\ \hline 1011 \\ \hline 20000 \end{array}$	г) $\begin{array}{r} 765 \\ + 576 \\ \hline 677 \\ \hline 2462 \end{array}$	д) $\begin{array}{r} 9E \\ + 56 \\ \hline 79 \\ \hline 167 \end{array}$
--	---	--	---	---

4.19. Найдите те подстановки десятичных цифр вместо букв, которые делают правильными выписанные результаты (разные цифры замещаются разными буквами):

а) $\begin{array}{r} ABCD \\ + ABCD \\ \hline BDCEC \end{array}$	б) $\begin{array}{r} A \\ + AB \\ \hline ABC \\ \hline BCB \end{array}$	
в) $\begin{array}{r} ABCDA \\ + FLCDA \\ \hline FLCLMN \end{array}$	г) $\begin{array}{r} ABCD \\ + EFBCA \\ \hline GHGCIJ \end{array}$	д) $\begin{array}{r} ABCD \\ + ABCE \\ \hline EGDHIG \end{array}$

4.20. Вычитите:

- | | | |
|-------------------------------|-----------------------------|--------------------------------|
| а) 111_2 из 10100_2 ; | д) 15_8 из 20_8 ; | и) $1A_{16}$ из 31_{16} ; |
| б) $10,11_2$ из $100,1_2$; | е) 47_8 из 102_8 ; | к) $F9E_{16}$ из $2A30_{16}$; |
| в) $111,1_2$ из 10010_2 ; | ж) $56,7_8$ из 101_8 ; | л) $D,1_{16}$ из $B,92_{16}$; |
| г) 10001_2 из $1110,11_2$; | з) $16,54_8$ из $30,01_8$; | м) ABC_{16} из 5678_{16} . |

4.21. Перемножьте числа, а затем проверьте результаты, выполнив соответствующие десятичные умножения:

- | | |
|------------------------------|--------------------------|
| а) 101101_2 и 101_2 ; | д) 37_8 и 4_8 ; |
| б) 111101_2 и $11,01_2$; | е) 16_8 и 7_8 ; |
| в) $1011,11_2$ и $101,1_2$; | ж) $7,5_8$ и $1,6_8$; |
| г) 101_2 и $1111,001_2$; | з) $6,25_8$ и $7,12_8$. |

4.22. Разделите 10010110_2 на 1010_2 и проверьте результат, умножая делитель на частное.

4.23. Разделите 10011010100_2 на 1100_2 и затем выполните соответствующее десятичное и восьмеричное деление.

4.24. Вычислите значения выражений:

- а) $256_8 + 10110,1_2 \cdot (60_8 + 12_{10}) - 1F_{16}$;
- б) $1AD_{16} - 100101100_2 : 1010_2 + 217_8$;
- в) $1010_{10} + (106_{16} - 11011101_2) \square 12_8$;
- г) $1011_2 \cdot 1100_2 : 14_8 + (100000_2 - 40_8)$.

4.25. Расположите следующие числа в порядке возрастания:

- а) $74_8, 110010_2, 70_{10}, 38_{16}$;
- б) $6E_{16}, 142_8, 1101001_2, 100_{10}$;
- в) $777_8, 101111111_2, 2FF_{16}, 500_{10}$;
- г) $100_{10}, 1100000_2, 60_{16}, 141_8$.

4.26. Запишите уменьшающийся ряд чисел $+3, +2, \dots, -3$ в однобайтовом формате:

- а) в прямом коде;
- б) в обратном коде;
- в) в дополнительном коде.

4.27. Запишите числа в прямом коде (формат 1 байт):

а) 31; б) -63; в) 65; г) -128.

4.28. Запишите числа в обратном и дополнительном кодах (формат 1 байт):

а) -9; б) -15; в) -127; г) -128.

4.29. Найдите десятичные представления чисел, записанных в дополнительном коде:

а) 1 1111000; б) 1 0011011; в) 1 1101001; г) 1 0000000.

4.30. Найдите десятичные представления чисел, записанных в обратном коде:

а) 1 1101000; б) 1 0011111; в) 1 0101011; г) 1 0000000.

4.31. Выполните вычитания чисел путем сложения их обратных (дополнительных) кодов в формате 1 байт. Укажите, в каких случаях имеет место переполнение разрядной сетки:

- а) 9 - 2; г) -20 - 10; ж) -120 - 15;
б) 2 - 9; д) 50 - 25; з) -126 - 1;
в) -5 - 7; е) 127 - 1; и) -127 - 1.
-
-

Ответы — Раздел 4. Арифметические основы компьютеров

4.1. в) троичная: 0, 1, 2, 10, 11, 12, 20, 21, 22, 100, 101, 102, 110, 111, 112, 120, 121, 122, 200, 201; **г)** пятеричная: 0, 1, 2, 3, 4, 10, 11, 12, 13, 14, 20, 21, 22, 23, 24, 30, 31, 32, 33, 34.

4.2. а) 10_2 ; **б)** 110_2 ; **в)** 1000_2 ; **г)** 10000_2 ; **д)** 101100_2 ; **е)** 2_8 ; **ж)** 10_8 ; **з)** 40_8 ; **и)** 200_8 ; **к)** 10000_8 ; **л)** 10_{16} ; **м)** 20_{16} ; **н)** 100_{16} ; **о)** $9AFA_{16}$; **п)** $CDF0_{16}$.

4.3. а) 1_2 ; **б)** 1001_2 ; **в)** 111_2 ; **г)** 1111_2 ; **д)** 10011_2 ; **е)** 7_8 ; **ж)** 17_8 ; **з)** 77_8 ; **и)** 107_8 ; **к)** 777_8 ; **л)** F_{16} ; **м)** $1F_{16}$; **н)** FF_{16} ; **о)** $A0F_{16}$; **п)** FFF_{16} .

4.4. Четное двоичное число оканчивается цифрой 0, нечетное двоичное — цифрой 1, четное троичное — цифрами 0, 1 или 2.

4.5. а) 7; **б)** 511; **в)** 4091.

4.7. а) ни в какой; **б)** в шестеричной.

4.8. Основание 5.

4.9. а) 91; **б)** 183; **в)** 225; **г)** $35/64$; **д)** 52,75; **е)** 335; **ж)** 520; **з)** 668; **и)** $7/16$; **к)** $83^{33}/64$; **л)** 31; **м)** 2748; **н)** 4112; **о)** $41/64$; **п)** $478^{25}/32$.

4.10. **a)** 1111101_2 ; 175_8 ; $7D_{16}$; **б)** 11100101_2 ; 345_8 ; $E5_{16}$; **в)** 1011000_2 ; 130_8 ; 58_{16} ; **г)** $100101,01_2$; $45,2_8$; $25,4_{16}$; **д)** $11001110,001_2$; $316,1_8$; $CE,2_{16}$.

4.11. **a)** $11767,34_8$; $13F7,7_{16}$; **б)** $1653,564_8$; $3AB,BA_{16}$; **в)** $271,547_8$; $B9,B38_{16}$; **г)** $13634,6_8$; $179C,C_{16}$; **д)** $27,7674_8$; $17,FBC_{16}$; **е)** $1425,62_8$; $315,C8_{16}$.

4.12. **a)** 1011001110_2 ; 1316_8 ; **б)** 1001111101000000_2 ; 117500_8 ; **в)** 10101011110011011110_2 ; 2536336_8 ; **г)** $100000010000,000100000001_2$; $10020,0401_8$; **д)** $1101010111100,10011101_2$; $15274,472_8$.

4.13. **a)** 101101_2 , 101110_2 , 101111_2 , 110000_2 ; **б)** 202_3 , 210_3 , 211_3 , 212_3 , 220_3 , 221_3 , 222_3 , 1000_3 ; **в)** 14_8 , 15_8 , 16_8 , 17_8 , 20_8 ; **г)** 28_{16} , 29_{16} , $2A_{16}$, $2B_{16}$, $2C_{16}$, $2D_{16}$, $2E_{16}$, $2F_{16}$, 30_{16} ;

4.14. **a)** $47_{10} - 101111_2 - 57_8 - 47_{10} - 57_8 - 101111_2 - 2F_{16} - 47_{10} - 2F_{16} - 101111_2 - 47_{10}$; **б)** $79_{10} - 1001111_2 - 117_8 - 79_{10} - 117_8 - 1001111_2 - 4F_{16} - 79_{10} - 4F_{16} - 1001111_2 - 79_{10}$.

4.15.

					+	0	1	2	3	4
					0	0	1	2	3	4
	+	0	1	2	1	1	2	3	4	10
	0	0	1	2	2	2	3	4	10	11
	1	1	2	10	3	3	4	10	11	12
	2	2	10	11	4	4	10	11	12	13

4.16.

					x	0	1	2	3	4
					0	0	0	0	0	0
	x	0	1	2	1	0	1	2	3	4
	0	0	0	0	2	0	2	4	11	13
	1	0	1	2	3	0	3	11	14	22
	2	0	2	11	4	0	4	13	22	31

4.17. **a)** 11010100_2 ; **б)** $10001,0_2$; **в)** $10101,1_2$; **г)** $11001,1_2$; **д)** 134_8 ; **е)** 224_8 ; **ж)** $24,3_8$; **з)** 34_8 ; **и)** 19_{16} ; **к)** 25_{16} ; **л)** $19,A_{16}$; **м)** 26_{16} .

4.18. а) в 16-й; **б)** в 10-й; **в)** в 3-й; **г)** в 8-й; **д)** в 16-й.

4.19. в) A=9, B=4, C=5, D=3, F=1, L=0, M=7, N=8; **г)** A=3, B=6, C=2, D=5, E=9, F=7, G=1, H=0, I=4, J=8; **д)** A=9, B=3, C=4, D=2, E=1, F=8, G=0, H=7, I=6.

4.20. а) 1101_2 ; **б)** $1,11_2$; **в)** $1010,1_2$; **г)** $-10,01_2$; **д)** 3_8 ; **е)** 33_8 ; **ж)** $22,1_8$; **з)** $11,25_8$; **и)** 17_{16} ; **к)** $1A92_{16}$; **л)** $-1,7E_{16}$; **м)** $4BBC_{16}$.

4.21. а) 11100001_2 ; **б)** $11000110,01_2$; **в)** $1000000,101_2$; **г)** $1001011,101_2$; **д)** 174_8 ; **е)** 142_8 ; **ж)** $15,26_8$; **з)** $55,2222_8$.

4.22. 1111_2 .

4.23. 1100111_2 ; 103_{10} ; 147_8 .

4.24. а) 1493_{10} ; **б)** 542_{10} ; **в)** 1420_{10} ; **г)** 11_{10} .

4.25. а) 110010_2 , 38_{16} , 74_8 , 70_{10} ; **б)** 142_8 , 100_{10} , 1101001_2 , $6E_{16}$; **в)** 10111111_2 , 500_{10} , 777_8 , $2FF_{16}$; **г)** 1100000_2 , 60_{16} , 141_8 , 100_{10} .

4.26. а) 00000011 , 00000010 , 00000001 , 00000000 , 10000001 , 10000010 , 10000011 ; **б)** 00000011 , 00000010 , 00000001 , 00000000 , 11111110 , 11111101 , 11111100 ; **в)** 00000011 , 00000010 , 00000001 , 00000000 , 11111111 , 11111110 , 11111101 .

4.27. а) 00001111 ; **б)** 10111111 ; **в)** 01000001 ; **г)** невозможно.

4.28. Обратный: **а)** 11110110 , **б)** 11110000 , **в)** 10000000 , **г)** невозможно.
Дополнительный: **а)** 11110111 ; **б)** 11110001 ; **в)** 10000001 ; **г)** 10000000 .

4.29. а) -8; б) -101; в) -23; г) -128.

4.30. а) -23; б) -96; в) -84; г) -127.

4.31. Обратный: а) 00000111; б) 11111000; в) 11110011; г) 11100001; д) 00011001; е) 01111110; ж) переполнение; з) 10000000; и) невозможно. Дополнительный: а) 00000111; б) 11111001; в) 11110100; г) 11100010; д) 00011001; е) 01111110; ж) переполнение; з) 10000001; и) 10000000.

Глава 5. Логические основы компьютеров

5.1. Что такое алгебра логики?

Алгебра логики — это раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

Алгебра логики возникла в середине XIX века в трудах английского математика **Джорджа Буля**. Ее создание представляло собой попытку решать традиционные логические задачи алгебраическими методами.

Что же такое логическое высказывание?

Логическое высказывание — это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.



Джордж Буль

Так, например, предложение "*6 — четное число*" следует считать высказыванием, так как оно истинное. Предложение "*Рим — столица Франции*" тоже высказывание, так как оно ложное.

Разумеется, **не всякое предложение является логическим высказыванием**. Высказываниями не являются, например, предложения "*ученик десятого класса*" и "*информатика — интересный предмет*". Первое предложение ничего не утверждает об ученике, а второе использует слишком неопределённое понятие "*интересный предмет*". Вопросительные и восклицательные предложения также не являются высказываниями, поскольку говорить об их истинности или ложности не имеет смысла.

Предложения типа "*в городе А более миллиона жителей*", "*у него голубые глаза*" не являются высказываниями, так как для выяснения их истинности или ложности нужны дополнительные сведения: о каком конкретно городе или человеке идет речь. Такие предложения называются *высказывательными формами*.

Высказывательная форма — это повествовательное предложение, которое прямо или косвенно содержит хотя бы одну переменную и становится высказыванием, когда все переменные замещаются своими значениями.

Алгебра логики рассматривает любое высказывание только с одной точки зрения — является ли оно истинным или ложным. Заметим, что **зачастую трудно установить истинность высказывания**. Так, например, высказывание "*площадь поверхности Индийского океана равна 75 млн кв. км*" в одной ситуации можно посчитать ложным, а в другой — истинным. Ложным — так как указанное значение неточное и вообще не является постоянным. Истинным — если рассматривать его как некоторое приближение, приемлемое на практике.

Употребляемые в обычной речи слова и словосочетания "**не**", "**и**", "**или**", "**если... , то**", "**тогда и только тогда**" и другие позволяют из уже заданных высказываний строить новые высказывания. Такие слова и словосочетания называются **логическими связками**.

Высказывания, образованные из других высказываний с помощью логических связок, называются **составными**. Высказывания, не являющиеся составными, называются **элементарными**.

Так, например, из элементарных высказываний "*Петров — врач*", "*Петров — шахматист*" при помощи связки "**и**" можно получить составное высказывание "*Петров — врач и шахматист*", понимаемое как "*Петров — врач, хорошо играющий в шахматы*".

При помощи связки "**или**" из этих же высказываний можно получить составное высказывание "*Петров — врач или шахматист*", понимаемое в алгебре логики как "*Петров или врач, или шахматист, или и врач и шахматист одновременно*".

Истинность или ложность получаемых таким образом составных высказываний зависит от истинности или ложности элементарных высказываний.

Чтобы обращаться к логическим высказываниям, им назначают имена. Пусть через **A** обозначено высказывание "*Тимур поедет летом на море*", а через **B** — высказывание "*Тимур летом отправится в горы*". Тогда составное высказывание "*Тимур летом побывает и на море, и в горах*" можно кратко записать как **A и B**. Здесь "**и**" — логическая связка, **A**, **B** — логические переменные, которые могут принимать только два значения — "истина" или "ложь", обозначаемые, соответственно, "1" и "0".

Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначение:

НЕ Операция, выражаемая словом "**не**", называется **отрицанием** и обозначается чертой над высказыванием (или знаком \neg). Высказывание \bar{A} истинно, когда **A** ложно, и ложно, когда **A** истинно. Пример. "*Луна — спутник Земли*" (**A**); "*Луна — не спутник Земли*" (\bar{A}).

И Операция, выражаемая связкой "**и**", называется **конъюнкцией** (лат. conjunctio — соединение) или логическим умножением и обозначается точкой "**·**" (может также обозначаться знаками **∧** или **&**). Высказывание **A · B** истинно тогда и только тогда, когда оба высказывания **A** и **B** истинны. Например, высказывание "*10 делится на 2 и 5 больше 3*" истинно, а высказывания "*10 делится на 2 и 5 не больше 3*", "*10 не делится на 2 и 5 больше 3*", "*10 не делится на 2 и 5 не больше 3*" — ложны.

ИЛИ Операция, выражаемая связкой "или" (в неисключающем смысле этого слова), называется **дизъюнкцией** (лат. *disjunctio* — разделение) или логическим сложением и обозначается знаком \vee (или плюсом). Высказывание $A \vee B$ ложно тогда и только тогда, когда оба высказывания A и B ложны. Например, высказывание "10 не делится на 2 или 5 не больше 3" ложно, а высказывания "10 делится на 2 или 5 больше 3", "10 делится на 2 или 5 не больше 3", "10 не делится на 2 или 5 больше 3" — истинны.

ЕСЛИ-ТО Операция, выражаемая связками "если ..., то", "из ... следует", "... влечет ...", называется **импликацией** (лат. *implico* — тесно связаны) и обозначается знаком \rightarrow . Высказывание $A \rightarrow B$ ложно тогда и только тогда, когда A истинно, а B ложно.

Каким же образом импликация связывает два элементарных высказывания? Покажем это на примере высказываний: "данный четырёхугольник — квадрат" (A) и "около данного четырёхугольника можно описать окружность" (B). Рассмотрим составное высказывание $A \rightarrow B$, понимаемое как "если данный четырёхугольник квадрат, то около него можно описать окружность". Есть **три варианта**, когда высказывание $A \rightarrow B$ истинно:

1. A истинно и B истинно, то есть данный четырёхугольник квадрат, и около него можно описать окружность;
2. A ложно и B истинно, то есть данный четырёхугольник не является квадратом, но около него можно описать окружность (разумеется, это справедливо не для всякого четырёхугольника);
3. A ложно и B ложно, то есть данный четырёхугольник не является квадратом, и около него нельзя описать окружность.

Ложен только один вариант, когда A истинно, а B ложно, то есть данный четырёхугольник является квадратом, но около него нельзя описать окружность.

В обычной речи связка "если ..., то" описывает причинно-следственную связь между высказываниями. Но в логических операциях смысл высказываний не учитывается. Рассматривается только их истинность или ложность. Поэтому не надо смущаться "бессмысленностью" импликаций, образованных высказываниями, совершенно не связанными по содержанию. Например, такими: "если президент США — демократ, то в Африке водятся жирафы", "если арбуз — ягода, то в бензоколонке есть бензин".

РАВНОСИЛЬНО Операция, выражаемая связками "тогда и только тогда", "необходимо и достаточно", "... равносильно ...", называется **эквиваленцией** или двойной импликацией и обозначается знаком \Leftrightarrow или \sim . Высказывание $A \Leftrightarrow B$ истинно тогда и только тогда, когда значения A и B совпадают. Например, высказывания "24 делится на 6 тогда и только тогда, когда 24 делится на 3", "23 делится на 6 тогда и только тогда, когда 23 делится на 3" истинны, а высказывания "24 делится на 6 тогда и только тогда, когда 24 делится на 5", "21 делится на 6 тогда и только тогда, когда 21 делится на 3" ложны.

Высказывания A и B , образующие составное высказывание $A \Leftrightarrow B$, могут быть совершенно не связаны по содержанию, например: "три больше двух" (A), "пингвины живут в Антарктиде" (B). Отрицаниями этих высказываний являются высказывания

"три не больше двух" (\overline{A}), "пингвины не живут в Антарктиде" (\overline{B}). Образованные из высказываний A и B составные высказывания $A \leftrightarrow B$ и $\overline{A} \leftrightarrow \overline{B}$ истинны, а высказывания $A \leftrightarrow \overline{B}$ и $\overline{A} \leftrightarrow B$ — ложны.

Итак, нами рассмотрены пять логических операций: отрицание, конъюнкция, дизъюнкция, импликация и эквиваленция.

Импликацию можно выразить через дизъюнкцию и отрицание:

$$A \rightarrow B = \overline{A} \vee B.$$

Эквиваленцию можно выразить через отрицание, дизъюнкцию и конъюнкцию:

$$A \leftrightarrow B = (\overline{A} \vee B) \cdot (\overline{B} \vee A).$$

Таким образом, операций отрицания, дизъюнкции и конъюнкции достаточно, чтобы описывать и обрабатывать логические высказывания.

Порядок выполнения логических операций задается круглыми скобками. Но для уменьшения числа скобок договорились считать, что сначала выполняется операция отрицания ("не"), затем конъюнкция ("и"), после конъюнкции — дизъюнкция ("или") и в последнюю очередь — импликация.

5.2. Что такое логическая формула?

С помощью логических переменных и символов логических операций любое высказывание можно формализовать, то есть заменить логической формулой.

Определение логической формулы:

1. Всякая логическая переменная и символы "истина" ("1") и "ложь" ("0") — формулы.
2. Если A и B — формулы, то \overline{A} , $A \cdot B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$ — формулы.
3. Никаких других формул в алгебре логики нет.

В п. 1 определены элементарные формулы; в п. 2 даны правила образования из любых данных формул новых формул.

В качестве примера рассмотрим высказывание "если я куплю яблоки или абрикосы, то приготовлю фруктовый пирог". Это высказывание формализуется в виде $(A \vee B) \rightarrow C$. Такая же формула соответствует высказыванию "если Игорь знает английский или японский язык, то он получит место переводчика".

Как показывает анализ формулы $(A \vee B) \rightarrow C$, при определённых сочетаниях значений переменных A , B и C она принимает значение "истина", а при некоторых других сочетаниях — значение "ложь" (разберите самостоятельно эти случаи). Такие формулы называются **выполнимыми**.

Некоторые формулы принимают значение "истина" при любых значениях истинности входящих в них переменных. Таковой будет, например, формула $A \vee \bar{A}$, соответствующая высказыванию "Этот треугольник прямоугольный или тупоугольный". Эта формула истинна и тогда, когда треугольник прямоугольный, и тогда, когда треугольник не прямоугольный. Такие формулы называются **тождественно истинными формулами** или **тавтологиями**. Высказывания, которые формализуются тавтологиями, называются **логически истинными высказываниями**.

В качестве другого примера рассмотрим формулу $A \cdot \bar{A}$, которой соответствует, например, высказывание "Катя самая высокая девочка в классе, и в классе есть девочки выше Кати". Очевидно, что эта формула ложна, так как либо A , либо \bar{A} обязательно ложно. Такие формулы называются **тождественно ложными формулами** или **противоречиями**. Высказывания, которые формализуются противоречиями, называются **логически ложными высказываниями**.

Если две формулы A и B одновременно, то есть при одинаковых наборах значений входящих в них переменных, принимают одинаковые значения, то они называются **равносильными**.

Равносильность двух формул алгебры логики обозначается символом "=" или символом "≡". Замена формулы другой, ей равносильной, называется **равносильным преобразованием** данной формулы.

5.3. Какая связь между алгеброй логики и двоичным кодированием?

Математический аппарат алгебры логики очень удобен для описания того, как функционируют аппаратные средства компьютера, поскольку основной системой счисления в компьютере является двоичная, в которой используются цифры 1 и 0, а значений логических переменных тоже два: "1" и "0".

Из этого следует два вывода:

1. одни и те же устройства компьютера могут применяться для обработки и хранения как числовой информации, представленной в двоичной системе счисления, так и логических переменных;
2. на этапе конструирования аппаратных средств алгебра логики позволяет значительно упростить логические функции, описывающие функционирование схем компьютера, и, следовательно, уменьшить число элементарных логических элементов, из десятков тысяч которых состоят основные узлы компьютера.

5.4. В каком виде записываются в памяти компьютера и в регистрах процессора данные и команды?

Данные и команды представляются в виде двоичных последовательностей различной структуры и длины. Существуют различные физические способы кодирования двоичной информации. Мы уже рассмотрели способы записи двоичной информации на [магнитных дисках](#) и на [CD-ROM](#). В электронных устройствах компьютера двоичные единицы чаще всего кодируются более высоким уровнем напряжения, чем двоичные нули (или наоборот), например:



5.5. Что такое логический элемент компьютера?

Логический элемент компьютера — это часть электронной логической схемы, которая реализует элементарную логическую функцию.

Логическими элементами компьютеров являются электронные схемы И, ИЛИ, НЕ, И—НЕ, ИЛИ—НЕ и другие (называемые также вентилями), а также триггер.

С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Обычно у вентиляей бывает от двух до восьми входов и один или два выхода.

Чтобы представить два логических состояния — “1” и “0” в вентилях, соответствующие им входные и выходные сигналы имеют один из двух установленных уровней напряжения. Например, +5 вольт и 0 вольт.

Высокий уровень обычно соответствует значению “истина” (“1”), а низкий — значению “ложь” (“0”).

Каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию, но не указывает на то, какая именно электронная схема в нем реализована. Это упрощает запись и понимание сложных логических схем.

Работу логических элементов описывают с помощью таблиц истинности.

Таблица истинности это табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

5.6. Что такое схемы И, ИЛИ, НЕ, И—НЕ, ИЛИ—НЕ?

С х е м а И

Схема И реализует конъюнкцию двух или более логических значений. Условное обозначение на структурных схемах схемы И с двумя входами представлено на рис. 5.1.

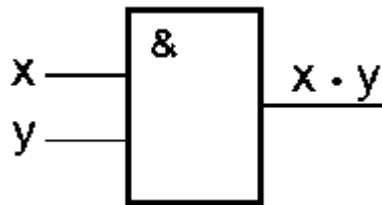


Рис. 5.1

Таблица истинности схемы И

x	y	x · y
0	0	0
0	1	0
1	0	0
1	1	1

Единица на выходе схемы И будет тогда и только тогда, когда на всех входах будут единицы. Когда хотя бы на одном входе будет ноль, на выходе также будет ноль.

Связь между выходом z этой схемы и входами x и y описывается соотношением:
 $z = x \cdot y$
 (читается как "x и y"). Операция конъюнкции на структурных схемах обозначается знаком "&" (читается как "амперсэнд"), являющимся сокращенной записью английского слова **and**.

С х е м а ИЛИ

Схема ИЛИ реализует дизъюнкцию двух или более логических значений. Когда хотя бы на одном входе схемы ИЛИ будет единица, на её выходе также будет единица.

Условное обозначение на структурных схемах схемы ИЛИ с двумя входами представлено на рис. 5.2. Знак "1" на схеме — от устаревшего обозначения дизъюнкции как " ≥ 1 " (т.е. значение дизъюнкции равно единице, если сумма значений операндов больше или равна 1). Связь между выходом z этой схемы и входами x и y описывается соотношением: $z = x \vee y$ (читается как "x или y").

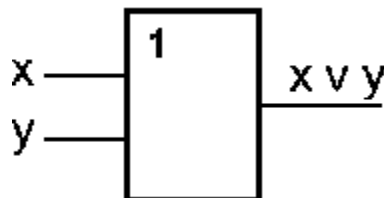


Рис. 5.2

Таблица истинности схемы **ИЛИ**

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

С х е м а Н Е

Схема **НЕ** (инвертор) реализует операцию отрицания. Связь между входом x этой схемы и выходом z можно записать соотношением $z = \overline{x}$, где \overline{x} читается как "не x " или "инверсия x ".

Если на входе схемы **0**, то на выходе **1**. Когда на входе **1**, на выходе **0**. Условное обозначение на структурных схемах инвертора — на рисунке 5.3

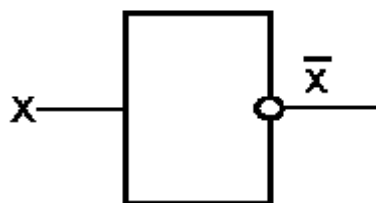


Рис. 5.3

Таблица истинности схемы **НЕ**

x	\overline{x}
0	1
1	0

С х е м а И—Н Е

Схема **И—НЕ** состоит из элемента **И** и инвертора и осуществляет отрицание результата схемы **И**. Связь между выходом z и входами x и y схемы записывают следующим образом: $z = \overline{x \cdot y}$, где $\overline{x \cdot y}$ читается как "инверсия x и y ". Условное обозначение на структурных схемах схемы **И—НЕ** с двумя входами представлено на рисунке 5.4.

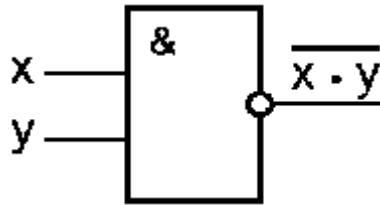


Рис. 5.4

Таблица истинности схемы И—НЕ

x	y	$\overline{x \cdot y}$
0	0	1
0	1	1
1	0	1
1	1	0

Схема ИЛИ—НЕ

Схема **ИЛИ—НЕ** состоит из элемента **ИЛИ** и инвертора и осуществляет отрицание результата схемы **ИЛИ**. Связь между выходом z и входами x и y схемы записывают следующим образом: $z = \overline{x \vee y}$, где $\overline{x \vee y}$, читается как "инверсия x или y ". Условное обозначение на структурных схемах схемы **ИЛИ—НЕ** с двумя входами представлено на рис. 5.5.

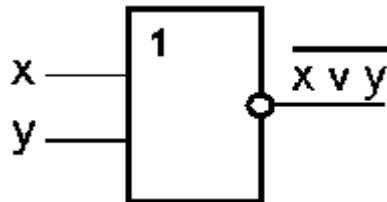


Рис. 5.5

Таблица истинности схемы ИЛИ—НЕ

x	y	$\overline{x \vee y}$
0	0	1
0	1	0
1	0	0
1	1	0

5.7. Что такое триггер?

Триггер — это электронная схема, широко применяемая в регистрах компьютера для надёжного запоминания одного разряда двоичного кода. Триггер имеет два устойчивых состояния, одно из которых соответствует двоичной единице, а другое — двоичному нулю.

Термин **триггер** происходит от английского слова **trigger** — защёлка, спусковой крючок. Для обозначения этой схемы в английском языке чаще употребляется термин **flip-flop**, что в переводе означает “хлопанье”. Это звукоподражательное название электронной схемы указывает на её способность почти мгновенно переходить (“перебрасываться”) из одного электрического состояния в другое и наоборот.

Самый распространённый тип триггера — так называемый RS-триггер (S и R, соответственно, от английских *set* — установка, и *reset* — сброс). Условное обозначение триггера — на рис. 5.6.

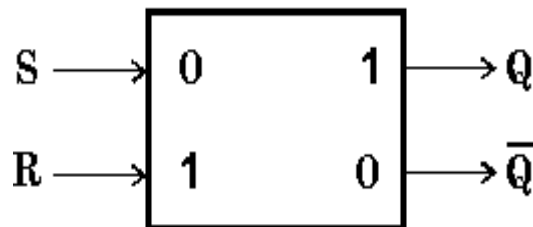



Рис. 5.6

Он имеет два симметричных входа S и R и два симметричных выхода Q и \bar{Q} , причем выходной сигнал Q является логическим отрицанием сигнала \bar{Q} .

На каждый из двух входов S и R могут подаваться входные сигналы в виде кратковременных импульсов ().

Наличие импульса на входе будем считать единицей, а его отсутствие — нулем.

На рис. 5.7 показана реализация триггера с помощью вентилях ИЛИ—НЕ и соответствующая таблица истинности.

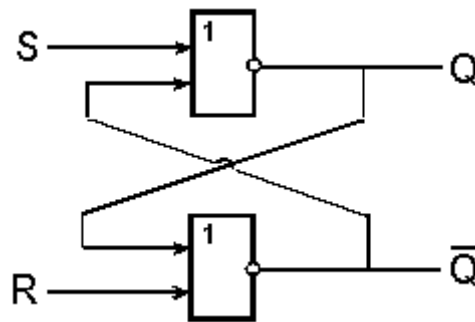


Рис. 5.7

S	R	Q	\bar{Q}
0	0	запрещено	
0	1	1	0
1	0	0	1
1	1	хранение бита	

Проанализируем возможные комбинации значений входов R и S триггера, используя его схему и таблицу истинности схемы ИЛИ—НЕ (табл. 5.5).

1. Если на входы триггера подать $S=“1”$, $R=“0”$, то (независимо от состояния) на выходе Q верхнего вентиля появится “0”. После этого на входах нижнего вентиля окажется $R=“0”$, $Q=“0”$ и выход \bar{Q} станет равным “1”.
2. Точно так же при подаче “0” на вход S и “1” на вход R на выходе \bar{Q} появится “0”, а на Q — “1”.
3. Если на входы R и S подана логическая “1”, то состояние Q и \bar{Q} не меняется.
4. Подача на оба входа R и S логического “0” может привести к неоднозначному результату, поэтому эта комбинация входных сигналов запрещена.

Поскольку один триггер может запомнить только один разряд двоичного кода, то для запоминания байта нужно 8 триггеров, для запоминания килобайта, соответственно, $8 \times 2^{10} = 8192$ триггеров. Современные микросхемы памяти содержат миллионы триггеров.

5.8. Что такое сумматор?

Сумматор — это электронная логическая схема, выполняющая суммирование двоичных чисел.

Сумматор служит, прежде всего, центральным узлом арифметико-логического устройства компьютера, однако он находит применение также и в других устройствах машины.

Многоразрядный двоичный сумматор, предназначенный для сложения многоразрядных двоичных чисел, представляет собой комбинацию **одноразрядных сумматоров**, с рассмотрения которых мы и начнём. Условное обозначение одноразрядного сумматора на рис. 5.8.

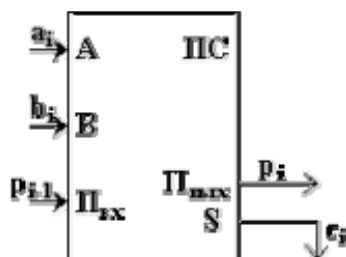


Рис. 5.8

При сложении чисел A и B в одном i -ом разряде приходится иметь дело с тремя цифрами:

1. цифра a_i первого слагаемого;
2. цифра b_i второго слагаемого;
3. перенос p_{i-1} из младшего разряда.

В результате сложения получаются две цифры:

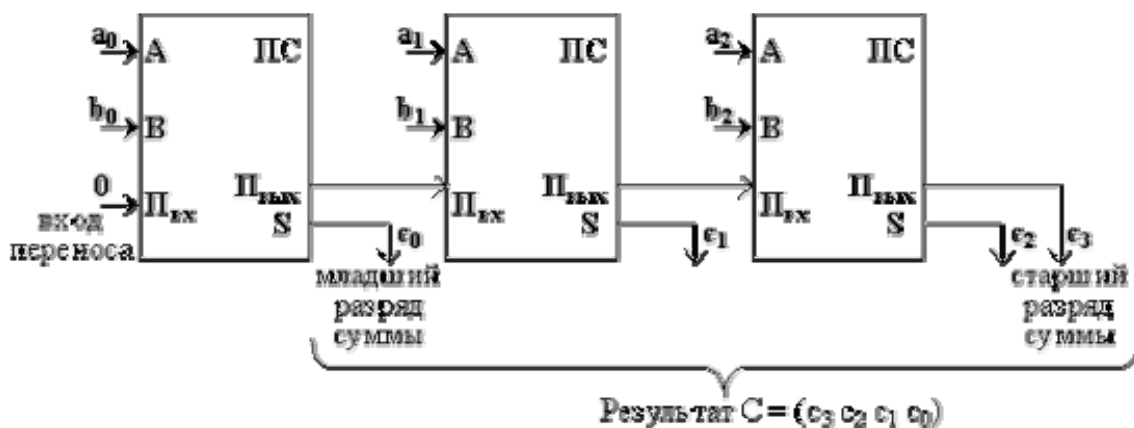
1. цифра c_i для суммы;
2. перенос p_i из данного разряда в старший.

Таким образом, **одноразрядный двоичный сумматор есть устройство с тремя входами и двумя выходами**, работа которого может быть описана следующей таблицей истинности:

Входы			Выходы	
Первое слагаемое	Второе слагаемое	Перенос	Сумма	Перенос
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Если требуется складывать двоичные слова длиной два и более бит, то можно использовать последовательное соединение таких сумматоров, причём **для двух соседних сумматоров выход переноса одного сумматора является входом для другого**.

Например, схема вычисления суммы $C = (c_3 c_2 c_1 c_0)$ двух двоичных трехразрядных чисел $A = (a_2 a_1 a_0)$ и $B = (b_2 b_1 b_0)$ может иметь вид:



5.9. Какие основные законы выполняются в алгебре логики?

В алгебре логики выполняются следующие основные законы, позволяющие производить тождественные преобразования логических выражений:

ОСНОВНЫЕ ЗАКОНЫ АЛГЕБРЫ ЛОГИКИ

Закон	Для ИЛИ	Для И
Переместительный	$x \vee y = y \vee x$	$x \cdot y = y \cdot x$
Сочетательный	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
Распределительный	$x \cdot (y \vee z) = x \cdot y \vee x \cdot z$	$x \vee (y \cdot z) = (x \vee y) \cdot (x \vee z)$
Правила де Моргана	$\overline{x \vee y} = \bar{x} \cdot \bar{y}$	$\overline{x \cdot y} = \bar{x} \vee \bar{y}$
Идемпотенции	$x \vee x = x$	$x \cdot x = x$
Поглощения	$x \vee (x \cdot y) = x$	$x \cdot (x \vee y) = x$
Склеивания	$(x \cdot y) \vee (\bar{x} \cdot y) = y$	$(x \vee y) \cdot (\bar{x} \vee y) = y$
Операция переменной с ее инверсией	$x \vee \bar{x} = 1$	$x \cdot \bar{x} = 0$
Операция с константами	$x \vee 0 = x; x \vee 1 = 1$	$x \cdot 1 = x; x \cdot 0 = 0$
Двойного отрицания	$\overline{\bar{x}} = x$	

5.10. Как составить таблицу истинности?

Согласно определению, таблица истинности логической формулы выражает соответствие между всевозможными наборами значений переменных и значениями формулы.

Для формулы, которая содержит две переменные, таких наборов значений переменных всего четыре:

$$(0, 0), \quad (0, 1), \quad (1, 0), \quad (1, 1).$$

Если формула содержит три переменные, то возможных наборов значений переменных восемь:

(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1).

Количество наборов для формулы с четырьмя переменными равно шестнадцати и т.д.

Удобной формой записи при нахождении значений формулы является таблица, содержащая кроме значений переменных и значений формулы также и значения промежуточных формул.

Примеры.

1. Составим таблицу истинности для формулы $\overline{x \cdot y \vee x \vee y \vee x}$, которая содержит две переменные x и y . В первых двух столбцах таблицы запишем четыре возможных пары значений этих переменных, в последующих столбцах — значения промежуточных формул и в последнем столбце — значение формулы. В результате получим таблицу:

Переменные		Промежуточные логические формулы					Формула
x	y	\bar{x}	$\bar{x} \cdot y$	$x \vee y$	$\overline{x \vee y}$	$x \cdot y \vee \overline{x \vee y}$	$\overline{x \cdot y \vee x \vee y \vee x}$
0	0	1	0	0	1	1	1
0	1	1	1	1	0	1	1
1	0	0	0	1	0	0	1
1	1	0	0	1	0	0	1

Из таблицы видно, что при всех наборах значений переменных x и y формула $\overline{x \cdot y \vee x \vee y \vee x}$ принимает значение 1, то есть является *тождественно истинной*.

2. Таблица истинности для формулы $\overline{x \vee y} \cdot (x \cdot \bar{y})$:

Переменные		Промежуточные логические формулы				Формула
x	y	$x \vee y$	$\overline{x \vee y}$	\bar{y}	$x \cdot \bar{y}$	$\overline{x \vee y} \cdot (x \cdot \bar{y})$
0	0	0	1	1	0	0
0	1	1	0	0	0	0
1	0	1	0	1	1	0
1	1	1	0	0	0	0

Из таблицы видно, что при всех наборах значений переменных x и y формула $\overline{x \vee y} \cdot (x \cdot \bar{y})$ принимает значение 0, то есть является *тождественно ложной*.

3. Таблица истинности для формулы $\overline{x \vee \overline{y}} \vee \overline{x} \cdot z$:

Переменные			Промежуточные логические формулы					Формула
x	y	z	\overline{y}	$x \vee \overline{y}$	$\overline{x \vee \overline{y}}$	\overline{x}	$\overline{x} \cdot z$	$\overline{x \vee \overline{y}} \vee \overline{x} \cdot z$
0	0	0	1	1	0	1	0	0
0	0	1	1	1	0	1	1	1
0	1	0	0	0	1	1	0	1
0	1	1	0	0	1	1	1	1
1	0	0	1	1	0	0	0	0
1	0	1	1	1	0	0	0	0
1	1	0	0	1	0	0	0	0
1	1	1	0	1	0	0	0	0

Из таблицы видно, что формула $\overline{x \vee \overline{y}} \vee \overline{x} \cdot z$ в некоторых случаях принимает значение 1, а в некоторых — 0, то есть является выполнимой.

5.11. Как упростить логическую формулу?

Равносильные преобразования логических формул имеют то же назначение, что и преобразования формул в обычной алгебре. Они служат для упрощения формул или приведения их к определённому виду путем использования основных законов алгебры логики.

Под упрощением формулы, не содержащей операций импликации и эквиваленции, понимают *равносильное преобразование*, приводящее к формуле, которая либо содержит по сравнению с исходной меньшее число операций конъюнкции и дизъюнкции и не содержит отрицаний неэлементарных формул, либо содержит меньшее число вхождений переменных.

Некоторые преобразования логических формул похожи на преобразования формул в обычной алгебре (вынесение общего множителя за скобки, использование переместительного и сочетательного законов и т.п.), тогда как другие преобразования основаны на свойствах, которыми не обладают операции обычной алгебры (использование распределительного закона для конъюнкции, законов поглощения, склеивания, де Моргана и др.).

Покажем на примерах некоторые приемы и способы, применяемые при упрощении логических формул:

$$1) \quad \overline{x \vee \overline{y}} \cdot (x \cdot \overline{y}) = \overline{x} \cdot \overline{y} \cdot (x \cdot \overline{y}) = \overline{x} \cdot x \cdot \overline{y} \cdot \overline{y} = 0 \cdot \overline{y} \cdot \overline{y} = 0 \cdot \overline{y} = 0$$

(законы алгебры логики применяются в следующей последовательности: правило де Моргана, сочетательный закон, правило операций переменной с её инверсией и правило операций с константами);

$$2) \quad \bar{x} \cdot y \vee \bar{x} \vee \bar{y} \vee x = \bar{x} \cdot y \vee \bar{x} \cdot \bar{y} \vee x = \bar{x} \cdot (y \vee \bar{y}) \vee x = \bar{x} \vee x = 1$$

(применяется правило де Моргана, выносятся за скобки общий множитель, используется правило операций переменной с её инверсией);

$$3) \quad (x \vee \bar{y}) \cdot (\bar{x} \vee y) \cdot (\bar{x} \vee \bar{y}) = (x \vee \bar{y}) \cdot (\bar{x} \vee y) \cdot (\bar{x} \vee \bar{y}) = y \cdot \bar{x}$$

(повторяется второй **сомножитель**, что разрешено законом идемпотенции; затем комбинируются два первых и два последних сомножителя и используется закон склеивания);

$$\begin{aligned} x \cdot \bar{y} \vee \bar{x} \cdot y \cdot z \vee x \cdot z &= x \cdot \bar{y} \vee \bar{x} \cdot y \cdot z \vee x \cdot z \cdot (y \vee \bar{y}) = \\ &= x \cdot \bar{y} \vee \bar{x} \cdot y \cdot z \vee x \cdot y \cdot z \vee x \cdot \bar{y} \cdot z = (x \cdot \bar{y} \vee x \cdot \bar{y} \cdot z) \vee (\bar{x} \cdot y \cdot z \vee x \cdot y \cdot z) = \\ 4) \quad &= x \cdot \bar{y} \vee y \cdot z \end{aligned}$$

(вводится **вспомогательный логический сомножитель** $(y \vee \bar{y})$; затем комбинируются два крайних и два средних логических слагаемых и используется закон поглощения);

$$5) \quad \overline{x \cdot y \vee z} = \overline{x \cdot y} \cdot \bar{z} = (\bar{x} \vee \bar{y}) \cdot \bar{z}$$

(сначала добиваемся, чтобы знак отрицания стоял только перед отдельными переменными, а не перед их комбинациями, для этого дважды применяем правило де Моргана; затем используем закон двойного отрицания);

$$6) \quad x \cdot y \vee x \cdot y \cdot z \vee x \cdot z \cdot p = x \cdot (y \cdot (1 \vee z) \vee z \cdot p) = x \cdot (y \vee z \cdot p)$$

(выносятся за скобки общие множители; применяется правило операций с константами);

$$\begin{aligned} x \vee \bar{y} \cdot \bar{z} \vee \bar{x} \vee y \vee \bar{z} &= x \vee \bar{y} \vee \bar{z} \vee \bar{x} \cdot \bar{y} \cdot \bar{z} = x \vee \bar{y} \vee z \vee x \cdot \bar{y} \cdot z = \\ 7) \quad &= x \vee z \vee (\bar{y} \vee x \cdot \bar{y} \cdot z) = x \vee z \vee \bar{y} \end{aligned}$$

(к отрицаниям неэлементарных формул применяется правило де Моргана; используются законы двойного отрицания и склеивания);

$$\begin{aligned} x \cdot \bar{y} \vee x \cdot y \cdot z \vee x \cdot \bar{y} \cdot z \vee x \cdot \bar{y} \cdot \bar{z} &= x \cdot (\bar{y} \vee y \cdot z \vee \bar{y} \cdot z \vee \bar{y} \cdot \bar{z}) = \\ 8) \quad &= x \cdot ((\bar{y} \vee \bar{y} \cdot z) \vee (y \cdot z \vee \bar{y} \cdot \bar{z})) = x \cdot (\bar{y} \vee \bar{y} \cdot z \vee 1) = x \cdot 1 = x \end{aligned}$$

(общий множитель x выносятся за скобки, комбинируются слагаемые в скобках — первое с третьим и второе с четвертым, к дизъюнкции $y \cdot z \vee \bar{y} \cdot \bar{z}$ применяется правило операции переменной с её инверсией);

$$\begin{aligned} (x \cdot \bar{y} \vee z) \cdot (\bar{x} \vee y) \vee \bar{z} &= x \cdot \bar{y} \cdot \bar{x} \vee x \cdot \bar{y} \cdot y \vee z \cdot \bar{x} \vee z \cdot y \vee \bar{z} = 0 \vee 0 \vee \\ &\vee z \cdot \bar{x} \vee z \cdot y \vee \bar{z} = z \cdot \bar{x} \vee (z \vee \bar{z}) \cdot (y \vee \bar{z}) = z \cdot \bar{x} \vee 1 \cdot (y \vee \bar{z}) = z \cdot \bar{x} \vee y \vee \\ &\vee \bar{z} = (z \cdot \bar{x} \vee \bar{z}) \vee y = (z \vee \bar{z}) \cdot (\bar{x} \vee \bar{z}) \vee y = 1 \cdot (\bar{x} \vee \bar{z}) \vee y = \bar{x} \vee \bar{z} \vee y \end{aligned}$$

9)

(используются распределительный закон для дизъюнкции, правило операции переменной с её инверсией, правило операций с константами, переместительный закон и распределительный закон для конъюнкции);

$$\begin{aligned}
 & x \cdot y \cdot (\bar{x} \cdot z \vee \overline{\bar{x} \cdot \bar{y} \cdot z} \vee z \cdot t) = x \cdot y \cdot (\bar{x} \cdot z \vee \overline{\bar{x} \cdot \bar{y} \cdot z} \vee z \cdot t) = \\
 10) & = x \cdot y \cdot (\bar{x} \cdot z \vee x \cdot y \vee \bar{z} \vee z \cdot t) = x \cdot y \vee x \cdot y \cdot \bar{z} \vee x \cdot y \cdot z \cdot t = x \cdot y
 \end{aligned}$$

(используются правило де Моргана, закон двойного отрицания и закон поглощения).

Из этих примеров видно, что при упрощении логических формул не всегда очевидно, какой из законов алгебры логики следует применить на том или ином шаге. Навыки приходят с опытом.

5.12. Что такое переключательная схема?

В компьютерах и других автоматических устройствах широко применяются электрические схемы, содержащие сотни и тысячи переключательных элементов: реле, выключателей и т.п. Разработка таких схем весьма трудоёмкое дело. Оказалось, что здесь с успехом может быть использован аппарат алгебры логики.

Переключательная схема — это схематическое изображение некоторого устройства, состоящего из переключателей и соединяющих их проводников, а также из входов и выходов, на которые подаётся и с которых снимается электрический сигнал.

Каждый переключатель имеет только два состояния: замкнутое и разомкнутое. Переключателю X поставим в соответствие логическую переменную x , которая принимает значение 1 в том и только в том случае, когда переключатель X замкнут и схема проводит ток; если же переключатель разомкнут, то x равен нулю.

Будем считать, что два переключателя X и \bar{X} связаны таким образом, что когда X замкнут, то \bar{X} разомкнут, и наоборот. Следовательно, если переключателю X поставлена в соответствие логическая переменная x , то переключателю \bar{X} должна соответствовать переменная \bar{x} .

Всей переключательной схеме также можно поставить в соответствие логическую переменную, равную единице, если схема проводит ток, и равную нулю — если не проводит. Эта переменная является функцией от переменных, соответствующих всем переключателям схемы, и называется **функцией проводимости**.

Найдем функции проводимости F некоторых переключательных схем:



Схема не содержит переключателей и проводит ток всегда, следовательно $F=1$;



Схема содержит один постоянно разомкнутый контакт, следовательно $F=0$;

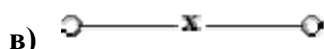


Схема проводит ток, когда переключатель x замкнут, и не проводит, когда x разомкнут, следовательно, $F(x) = x$;

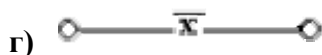


Схема проводит ток, когда переключатель x разомкнут, и не проводит, когда x замкнут, следовательно, $F(x) = \bar{x}$;

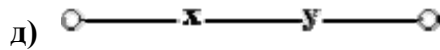


Схема проводит ток, когда оба переключателя замкнуты, следовательно, $F(x) = x \cdot y$;

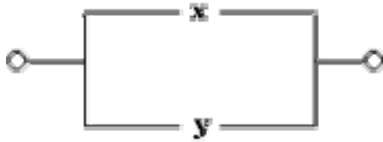


Схема проводит ток, когда хотя бы один из переключателей замкнут, следовательно, $F(x) = x \vee y$;

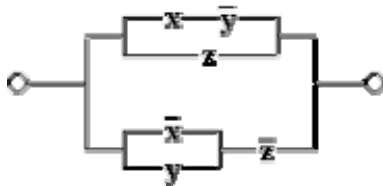


Схема состоит из двух параллельных ветвей и описывается функцией $F(x, y, z) = (x \cdot \bar{y}) \vee z \vee (\bar{x} \vee y) \cdot \bar{z}$.

Две схемы называются равносильными, если через одну из них проходит ток тогда и только тогда, когда он проходит через другую (при одном и том же входном сигнале).

Из двух равносильных схем более простой считается та схема, функция проводимости которой содержит меньшее число логических операций или переключателей.

Задача нахождения среди равносильных схем наиболее простых является очень важной. Большой вклад в ее решение внесли российские учёные **Ю.И. Журавлев, С.В. Яблонский** и др.

При рассмотрении переключательных схем возникают две основные задачи: **синтез и анализ схемы**.

СИНТЕЗ СХЕМЫ по заданным условиям ее работы сводится к следующим трём этапам:

1. составлению функции проводимости по таблице истинности, отражающей эти условия;
2. упрощению этой функции;
3. построению соответствующей схемы.

АНАЛИЗ СХЕМЫ сводится к

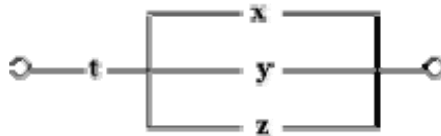
1. определению значений её функции проводимости при всех возможных наборах входящих в эту функцию переменных.

2. получению упрощённой формулы.

Примеры.

1. Построим схему, содержащую 4 переключателя x, y, z и t, такую, чтобы она проводила ток тогда и только тогда, когда замкнут контакт переключателя t и какой-нибудь из остальных трёх контактов.

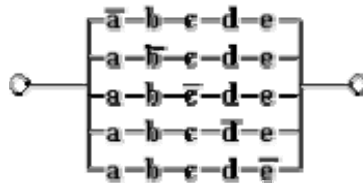
Решение. В этом случае можно обойтись без построения таблицы истинности. Очевидно, что функция проводимости имеет вид $F(x, y, z, t) = t \cdot (x \vee y \vee z)$, а схема выглядит так:



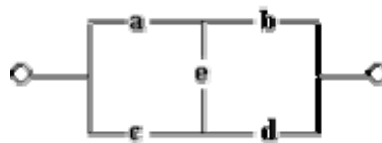
2. Построим схему с пятью переключателями, которая проводит ток в том и только в том случае, когда замкнуты ровно четыре из этих переключателей.

Решение: $F(a, b, c, d, e) = \bar{a} \cdot b \cdot c \cdot d \cdot e \vee a \cdot \bar{b} \cdot c \cdot d \cdot e \vee a \cdot b \cdot \bar{c} \cdot d \cdot e \vee a \cdot b \cdot c \cdot \bar{d} \cdot e \vee a \cdot b \cdot c \cdot d \cdot \bar{e}$

Схема имеет вид:

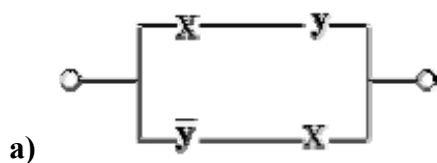


3. Найдем функцию проводимости схемы:

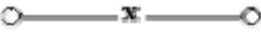


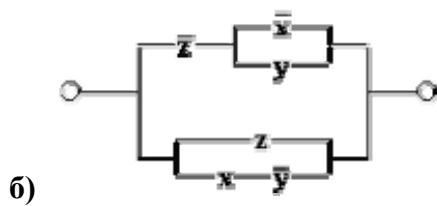
Решение. Имеется четыре возможных пути прохождения тока при замкнутых переключателях a, b, c, d, e : через переключатели a, b; через переключатели a, e, d; через переключатели c, d и через переключатели c, e, b. Функция проводимости $F(a, b, c, d, e) = a \cdot b \vee a \cdot e \cdot d \vee c \cdot d \vee c \cdot e \cdot b$.

4. Упростим переключательные схемы:




Решение: $F(x, y) = x \cdot y \vee \bar{y} \cdot x = x \cdot (y \vee \bar{y}) = x \cdot 1 = x$

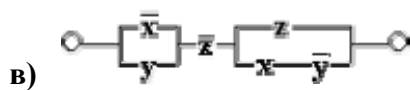
Упрощенная схема: 




Решение : $F(x, y, z) = \bar{z} \cdot (\bar{x} \vee y) \vee (z \vee x \cdot \bar{y}) = 1$

Здесь первое логическое слагаемое $\bar{z} \cdot (\bar{x} \vee y)$ является отрицанием второго логического слагаемого $(z \vee x \cdot \bar{y})$, а дизъюнкция переменной с ее инверсией равна 1.

Упрощенная схема : 

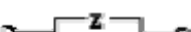


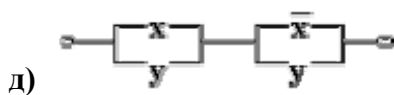
Решение : $F(x, y, z) = (\bar{x} \vee y) \cdot \bar{z} \cdot (z \vee x \cdot \bar{y}) = (\bar{x} \cdot \bar{z} \vee y \cdot \bar{z}) \cdot (z \vee x \cdot \bar{y}) = \bar{x} \cdot \bar{z} \cdot z \vee \bar{x} \cdot \bar{z} \cdot x \cdot \bar{y} \vee y \cdot \bar{z} \cdot z \vee y \cdot \bar{z} \cdot x \cdot \bar{y} = 0 \vee 0 \vee 0 \vee 0 = 0$

Упрощенная схема: 




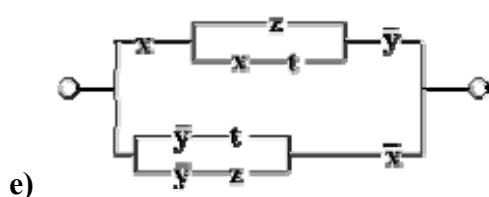
Решение : $F(x, y, z, t) = (x \vee \bar{y} \vee y) \cdot (z \vee t) = 1 \cdot (z \vee t) = z \vee t$

Упрощенная схема: 



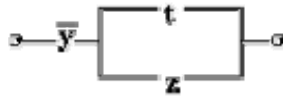
Решение : $F(x, y) = (x \vee y) \cdot (\bar{x} \vee y) = y$ (по закону склеивания)

Упрощенная схема: 



Решение:

$$\begin{aligned} F(x, y, z, t) &= x \cdot (z \vee x \cdot t) \cdot \bar{y} \vee (\bar{y} \cdot t \vee \bar{y} \cdot z) \cdot \bar{x} = x \cdot \bar{y} \cdot z \vee x \cdot \bar{y} \cdot x \cdot t \vee \bar{x} \cdot \bar{y} \cdot (t \vee z) = \\ &= x \cdot \bar{y} \cdot (z \vee t) \vee \bar{x} \cdot \bar{y} \cdot (t \vee z) = (t \vee z) \cdot (x \cdot \bar{y} \vee \bar{x} \cdot \bar{y}) = (t \vee z) \cdot \bar{y} \cdot (x \vee \bar{x}) = \bar{y} \cdot (t \vee z) \end{aligned}$$



Упрощенная схема:

5.13. Как решать логические задачи?

Разнообразие логических задач очень велико. Способов их решения тоже немало. Но наибольшее распространение получили следующие три способа решения логических задач:

- средствами алгебры логики;
- табличный;
- с помощью рассуждений.

Познакомимся с ними поочередно.

I. Решение логических задач средствами алгебры логики

Обычно используется следующая схема решения:

1. изучается условие задачи;
2. вводится система обозначений для логических высказываний;
3. конструируется логическая формула, описывающая логические связи между всеми высказываниями условия задачи;
4. определяются значения истинности этой логической формулы;
5. из полученных значений истинности формулы определяются значения истинности введенных логических высказываний, на основании которых делается заключение о решении.

Пример 1. Трое друзей, болельщиков автогонок "Формула-1", спорили о результатах предстоящего этапа гонок.

— Вот увидишь, Шумахер не придет первым, — сказал Джон. Первым будет Хилл.

— Да нет же, победителем будет, как всегда, Шумахер, — воскликнул Ник. — А об Алези и говорить нечего, ему не быть первым.

Питер, к которому обратился Ник, возмутился:

— Хиллу не видать первого места, а вот Алези пилотирует самую мощную машину.

По завершении этапа гонок оказалось, что каждое из двух предположений двоих друзей подтвердилось, а оба предположения третьего из друзей оказались неверны. Кто выиграл этап гонки?

Решение. Введем обозначения для логических высказываний:

Ш — победит Шумахер; **X** — победит Хилл; **A** — победит Алези.

Реплика Ника "Алези пилотирует самую мощную машину" не содержит никакого утверждения о месте, которое займёт этот гонщик, поэтому в дальнейших рассуждениях не учитывается.

Зафиксируем высказывания каждого из друзей:

Джон: $\overline{Ш} \cdot X$, **Ник:** $Ш \cdot \overline{A}$, **Питер:** \overline{X} .

Учитывая то, что предположения двух друзей подтвердились, а предположения третьего неверны, запишем и упростим истинное высказывание

$$(\overline{Ш} \cdot X) \cdot (Ш \cdot \overline{A}) \cdot \overline{X} \vee (\overline{Ш} \cdot X) \cdot (\overline{Ш} \cdot \overline{A}) \cdot \overline{X} \vee (\overline{Ш} \cdot X) \cdot (Ш \cdot \overline{A}) \cdot \overline{X} = (Ш \vee \overline{X}) \cdot Ш \cdot \overline{A} \cdot \overline{X} = Ш \cdot \overline{A} \cdot \overline{X}$$

Высказывание $Ш \cdot \overline{A} \cdot \overline{X}$ истинно только при $Ш=1, A=0, X=0$.

Ответ. Победителем этапа гонок стал Шумахер.

Пример 2. Некий любитель приключений отправился в кругосветное путешествие на яхте, оснащённой бортовым компьютером. Его предупредили, что чаще всего выходят из строя три узла компьютера — a, b, c , и дали необходимые детали для замены. Выяснить, какой именно узел надо заменить, он может по сигнальным лампочкам на контрольной панели. Лампочек тоже ровно три: x, y и z .

Инструкция по выявлению неисправных узлов такова:

1. если неисправен хотя бы один из узлов компьютера, то горит по крайней мере одна из лампочек x, y, z ;
2. если неисправен узел a , но исправен узел c , то загорается лампочка y ;
3. если неисправен узел c , но исправен узел b , загорается лампочка y , но не загорается лампочка x ;
4. если неисправен узел b , но исправен узел c , то загораются лампочки x и y или не загорается лампочка x ;
5. если горит лампочка x и при этом либо неисправен узел a , либо все три узла a, b, c исправны, то горит и лампочка y .

В пути компьютер сломался. На контрольной панели загорелась лампочка x . Тщательно изучив инструкцию, путешественник починил компьютер. Но с этого момента и до конца плавания его не оставляла тревога. Он понял, что инструкция несовершенна, и есть случаи, когда она ему не поможет.

Какие узлы заменил путешественник? Какие изъяны он обнаружил в инструкции?

Решение. Введем обозначения для логических высказываний:

a — неисправен узел a ; x — горит лампочка x ;

b — неисправен узел b ; y — горит лампочка y ;

c — неисправен узел c ; z — горит лампочка z .

Правила 1-5 выражаются следующими формулами:

$$a \vee b \vee c \rightarrow x \vee y \vee z \quad (1)$$

$$a \cdot \bar{c} \rightarrow y \quad (2)$$

$$c \cdot \bar{b} \rightarrow y \cdot \bar{x} \quad (3)$$

$$b \cdot \bar{c} \rightarrow (x \cdot y \vee \bar{x}) \quad (4)$$

$$(a \vee \bar{a} \cdot \bar{b} \cdot \bar{c}) \cdot x \rightarrow y \quad (5)$$

Формулы 1-5 истинны по условию, следовательно, их конъюнкция тоже истинна:

$$(a \vee b \vee c \rightarrow x \vee y \vee z) \cdot (a \cdot \bar{c} \rightarrow y) \cdot (c \cdot \bar{b} \rightarrow y \cdot \bar{x}) \cdot (b \cdot \bar{c} \rightarrow x \cdot y \vee \bar{x}) \cdot ((a \vee \bar{a} \cdot \bar{b} \cdot \bar{c}) \cdot x \rightarrow y) = 1$$

Выражая импликацию через дизъюнкцию и отрицание (напомним, что $(a \rightarrow b) = (\bar{a} \vee b)$), получаем:

$$\begin{aligned} & (\bar{a} \cdot \bar{b} \cdot \bar{c} \vee x \vee y \vee z) \cdot (\bar{a} \vee c \vee y) \cdot (\bar{c} \vee b \vee y \cdot \bar{x}) \cdot (\bar{b} \vee c \vee x \cdot y \vee \bar{x}) \cdot \overline{((a \vee \bar{a}) \cdot (a \vee \bar{b})) \cdot (a \vee \bar{c}) \cdot x \vee y} = \\ & = (\bar{a} \cdot \bar{b} \cdot \bar{c} \vee x \vee y \vee z) \cdot (\bar{a} \vee c \vee y) \cdot (\bar{c} \vee b \vee y \cdot \bar{x}) \cdot (\bar{b} \vee c \vee x \cdot y \vee \bar{x}) \cdot (\bar{a} \cdot b \vee \bar{a} \cdot c \vee \bar{x} \vee y) = 1 \end{aligned}$$

Подставляя в это тождество конкретные значения истинности $x=1, y=0, z=0$, получаем:

$$(\bar{a} \vee c) \cdot (\bar{c} \vee b) \cdot (\bar{b} \vee c) \cdot (\bar{a} \cdot b \vee \bar{a} \cdot c) = (\bar{a} \cdot \bar{c} \vee \bar{a} \cdot b \vee c \cdot b) \cdot (\bar{b} \cdot \bar{a} \cdot c \vee c \cdot \bar{a} \cdot b \vee \bar{a} \cdot c) = \bar{a} \cdot b \cdot c = 1$$

Отсюда следует, что $a=0, b=1, c=1$.

Ответ на первый вопрос задачи: нужно заменить блоки b и c ; блок a не требует замены. Ответ на второй вопрос задачи получите самостоятельно.

II. Решение логических задач табличным способом

При использовании этого способа условия, которые содержит задача, и результаты рассуждений фиксируются с помощью специально составленных таблиц.

Пример 3. В симфонический оркестр приняли на работу трёх музыкантов: Брауна, Смита и Вессона, умеющих играть на скрипке, флейте, альте, кларнете, гобое и трубе.

Известно, что:

1. Смит самый высокий;
2. играющий на скрипке меньше ростом играющего на флейте;
3. играющие на скрипке и флейте и Браун любят пиццу;
4. когда между альтистом и трубачом возникает ссора, Смит мирит их;
5. Браун не умеет играть ни на трубе, ни на гобое.

На каких инструментах играет каждый из музыкантов, если каждый владеет двумя инструментами?

Решение. Составим таблицу и отразим в ней условия задачи, заполнив соответствующие клетки цифрами 0 и 1 в зависимости от того, ложно или истинно соответствующее высказывание.

Так как музыкантов трое, инструментов шесть и каждый владеет только двумя инструментами, получается, что каждый музыкант играет на инструментах, которыми остальные не владеют.

Из условия 4 следует, что Смит не играет ни на альте, ни на трубе, а из условий 3 и 5, что Браун не умеет играть на скрипке, флейте, трубе и гобое. Следовательно, инструменты Брауна — альт и кларнет. Занесем это в таблицу, а оставшиеся клетки столбцов "альт" и "кларнет" заполним нулями:

	скрипка	флейта	альт	кларнет	гобой	труба
Браун	0	0	1	1	0	0
Смит			0	0		0
Вессон			0	0		

Из таблицы видно, что на трубе может играть только Вессон.

Из условий 1 и 2 следует, что Смит не скрипач. Так как на скрипке не играет ни Браун, ни Смит, то скрипачом является Вессон. Оба инструмента, на которых играет Вессон, теперь определены, поэтому остальные клетки строки "Вессон" можно заполнить нулями:

	скрипка	флейта	альт	кларнет	гобой	труба
Браун	0	0	1	1	0	0
Смит	0		0	0		0
Вессон	1	0	0	0	0	1

Из таблицы видно, что играть на флейте и на гобое может только Смит.

	скрипка	флейта	альт	кларнет	гобой	труба
Браун	0	0	1	1	0	0
Смит	0	1	0	0	1	0
Вессон	1	0	0	0	0	1

Ответ: Браун играет на альте и кларнете, Смит — на флейте и гобое, Вессон — на скрипке и трубе.

Пример 4. Три одноклассника — Влад, Тимур и Юра, встретились спустя 10 лет после окончания школы. Выяснилось, что один из них стал врачом, другой физиком, а третий юристом. Один полюбил туризм, другой бег, страсть третьего — регби.

Юра сказал, что на туризм ему не хватает времени, хотя его сестра — единственный врач в семье, заядлый турист. Врач сказал, что он разделяет увлечение коллеги.

Забавно, но у двоих из друзей в названиях их профессий и увлечений не встречается ни одна буква их имен.

Определите, кто чем любит заниматься в свободное время и у кого какая профессия.

Решение. Здесь исходные данные разбиваются на тройки (имя — профессия — увлечение).

Из слов Юры ясно, что он не увлекается туризмом и он не врач. Из слов врача следует, что он турист.

Имя	Юра		
Профессия		врач	
Увлечение		туризм	

Буква "а", присутствующая в слове "врач", указывает на то, что Влад тоже не врач, следовательно врач — Тимур. В его имени есть буквы "т" и "р", встречающиеся в слове "туризм", следовательно второй из друзей, в названиях профессии и увлечения которого не встречается ни одна буква его имени — Юра. Юра не юрист и не регбист, так как в его имени содержатся буквы "ю" и "р". Следовательно, окончательно имеем:

Имя	Юра	Тимур	Влад
Профессия	физик	врач	юрист
Увлечение	бег	туризм	регби

Ответ. Влад — юрист и регбист, Тимур — врач и турист, Юра — физик и бегун.

Пример 5. Три дочери писательницы Дорис Кей — Джуди, Айрис и Линда, тоже очень талантливы. Они приобрели известность в разных видах искусств — пении, балете и кино. Все они живут в разных городах, поэтому Дорис часто звонит им в Париж, Рим и Чикаго.

Известно, что:

1. Джуди живет не в Париже, а Линда — не в Риме;
2. парижанка не снимается в кино;
3. та, кто живет в Риме, певица;
4. Линда равнодушна к балету.

Где живет Айрис, и какова ее профессия?

Решение. Составим таблицу и отразим в ней условия 1 и 4, заполнив клетки цифрами 0 и 1 в зависимости от того, ложно или истинно соответствующее высказывание:

Париж	Рим	Чикаго		Пение	Балет	Кино
0			Джуди			
			Айрис			
	0		Линда		0	

Далее рассуждаем следующим образом. Так как Линда живет не в Риме, то, согласно условию 3, она не певица. В клетку, соответствующую строке "Линда" и столбцу "Пение", ставим 0.

Из таблицы сразу видно, что Линда киноактриса, а Джуди и Айрис не снимаются в кино.

Париж	Рим	Чикаго		Пение	Балет	Кино
0			Джуди			0
			Айрис			0
	0		Линда	0	0	1

Согласно условию 2, парижанка не снимается в кино, следовательно, Линда живет не в Париже. Но она живет и не в Риме. Следовательно, Линда живет в Чикаго. Так как Линда и Джуди живут не в Париже, там живет Айрис. Джуди живет в Риме и, согласно условию 3, является певицей. А так как Линда киноактриса, то Айрис балерина.

В результате постепенного заполнения получаем следующую таблицу:

Париж	Рим	Чикаго		Пение	Балет	Кино
0	0	1	Джуди	1	0	0
1	0	0	Айрис	0	1	0
0	0	1	Линда	0	0	1

Ответ. Айрис балерина. Она живет в Париже.

III. Решение логических задач с помощью рассуждений

Этим способом обычно решают несложные логические задачи.

Пример 6. Вадим, Сергей и Михаил изучают различные иностранные языки: китайский, японский и арабский. На вопрос, какой язык изучает каждый из них, один ответил: "Вадим изучает китайский, Сергей не изучает китайский, а Михаил не изучает арабский". Впоследствии выяснилось, что в этом ответе только одно утверждение верно, а два других ложны. Какой язык изучает каждый из молодых людей?

Решение. Имеется три утверждения:

1. Вадим изучает китайский;
2. Сергей не изучает китайский;
3. Михаил не изучает арабский.

Если верно первое утверждение, то верно и второе, так как юноши изучают разные языки. Это противоречит условию задачи, поэтому первое утверждение ложно.

Если верно второе утверждение, то первое и третье должны быть ложны. При этом получается, что никто не изучает китайский. Это противоречит условию, поэтому второе утверждение тоже ложно.

Остается считать верным третье утверждение, а первое и второе — ложными. Следовательно, Вадим не изучает китайский, китайский изучает Сергей.

Ответ: Сергей изучает китайский язык, Михаил — японский, Вадим — арабский.

Пример 7. В поездке пятеро друзей — Антон, Борис, Вадим, Дима и Гриша, познакомились с попутчицей. Они предложили ей отгадать их фамилии, причём каждый из них высказал одно истинное и одно ложное утверждение:

Дима сказал: "Моя фамилия — Мишин, а фамилия Бориса — Хохлов". Антон сказал: "Мишин — это моя фамилия, а фамилия Вадима — Белкин". Борис сказал: "Фамилия Вадима — Тихонов, а моя фамилия — Мишин". Вадим сказал: "Моя фамилия — Белкин, а фамилия Гриши — Чехов". Гриша сказал: "Да, моя фамилия Чехов, а фамилия Антона — Тихонов".

Какую фамилию носит каждый из друзей?

Решение. Обозначим высказывательную форму "юноша по имени А носит фамилию Б" как A_B , где буквы А и Б соответствуют начальным буквам имени и фамилии.

Зафиксируем высказывания каждого из друзей:

1. D_M и B_X ;
2. A_M и V_B ;
3. V_T и B_M ;
4. V_B и G_C ;
5. G_C и A_T .

Допустим сначала, что истинно D_M . Но, если истинно D_M , то у Антона и у Бориса должны быть другие фамилии, значит A_M и B_M ложно. Но если A_M и B_M ложны, то должны быть истинны V_B и V_T , но V_B и V_T одновременно истинными быть не могут.

Значит остается другой случай: истинно B_X . Этот случай приводит к цепочке умозаключений:

B_X истинно $\rightarrow B_M$ ложно $\rightarrow V_T$ истинно $\rightarrow A_T$ ложно $\rightarrow G_C$ истинно $\rightarrow V_B$ ложно $\rightarrow A_M$ истинно.

Ответ: Борис — Хохлов, Вадим — Тихонов, Гриша — Чехов, Антон — Мишин, Дима — Белкин.

Пример 8. Министры иностранных дел России, США и Китая обсудили за закрытыми дверями проекты соглашения о полном разоружении, представленные каждой из стран. Отвечая затем на вопрос журналистов: "Чей именно проект был принят?", министры дали такие ответы:

Россия — "Проект не наш, проект не США";
США — "Проект не России, проект Китая";
Китай — "Проект не наш, проект России".

Один из них (самый откровенный) оба раза говорил правду; второй (самый скрытный) оба раза говорил неправду, третий (осторожный) один раз сказал правду, а другой раз — неправду.

Определите, представителями каких стран являются откровенный, скрытный и осторожный министры.

Решение. Для удобства записи пронумеруем высказывания дипломатов:

Россия — "Проект не наш" (1), "Проект не США" (2);
США — "Проект не России" (3), "Проект Китая" (4);
Китай — "Проект не наш" (5), "Проект России" (6).

Узнаем, кто из министров самый откровенный.

Если это российский министр, то из справедливости (1) и (2) следует, что победил китайский проект. Но тогда оба утверждения министра США тоже справедливы, чего не может быть по условию.

Если самый откровенный — министр США, то тогда вновь получаем, что победил китайский проект, значит оба утверждения российского министра тоже верны, чего не может быть по условию.

Получается, что наиболее откровенным был китайский министр. Действительно, из того, что (5) и (6) справедливы, следует, что победил российский проект. А тогда получается, что из двух утверждений российского министра первое ложно, а второе верно. Оба же утверждения министра США неверны.

Ответ: Откровеннее был китайский министр, осторожнее — российский, скрытнее — министр США.

5.14. Упражнения

5.1. Установите, какие из следующих предложений являются логическими высказываниями, а какие — нет (объясните почему):

- а) "Солнце есть спутник Земли";
- б) " $2+3=4$ ";

- **в)** "сегодня отличная погода";
- **г)** "в романе Л.Н. Толстого "Война и мир" 3 432 536 слов";
- **д)** "Санкт-Петербург расположен на Неве";
- **е)** "музыка Баха слишком сложна";
- **ж)** "первая космическая скорость равна 7.8 км/сек";
- **з)** "железо — металл";
- **и)** "если один угол в треугольнике прямой, то треугольник будет тупоугольным";
- **к)** "если сумма квадратов двух сторон треугольника равна квадрату третьей, то он прямоугольный".

5.2. Укажите, какие из высказываний предыдущего упражнения истинны, какие — ложны, а какие относятся к числу тех, истинность которых трудно или невозможно установить.

5.3. Приведите примеры истинных и ложных высказываний:

- **а)** из арифметики; **б)** из физики;
- **в)** из биологии; **г)** из информатики;
- **д)** из геометрии; **е)** из жизни.

5.4. Сформулируйте отрицания следующих высказываний или высказывательных форм:

- **а)** "Эльбрус — высочайшая горная вершина Европы";
- **б)** " $2 > 5$ ";
- **в)** " $10 < 7$ ";
- **г)** "все натуральные числа целые";
- **д)** "через любые три точки на плоскости можно провести окружность";
- **е)** "теннисист Кафельников не проиграл финальную игру";
- **ж)** "мишень поражена первым выстрелом";
- **з)** "это утро ясное и теплое";
- **и)** "число n делится на 2 или на 3";
- **к)** "этот треугольник равнобедренный и прямоугольный";
- **л)** "на контрольной работе каждый ученик писал своей ручкой".

5.5. Определите, какие из высказываний (высказывательных форм) в следующих парах являются отрицаниями друг друга, а какие нет:

- **а)** " $5 < 10$ ", " $5 > 10$ ";
- **б)** " $10 > 9$ ", " $10 \leq 9$ ";
- **в)** "мишень поражена первым выстрелом", "мишень поражена вторым выстрелом";
- **г)** "машина останавливалась у каждого из двух светофоров", "машина не останавливалась у каждого из двух светофоров";
- **д)** "человечеству известны все планеты Солнечной системы", "в Солнечной системе есть планеты, неизвестные человечеству";
- **е)** "существуют белые слоны", "все слоны серые";
- **ж)** "кит — млекопитающее", "кит — рыба";
- **з)** "неверно, что точка A не лежит на прямой a ", "точка A лежит на прямой a ";
- **и)** "прямая a параллельна прямой b ", "прямая a перпендикулярна прямой b ";

- **к)** "этот треугольник равнобедренный и прямоугольный", "этот треугольник не равнобедренный или он не прямоугольный".

5.6. Определите значения истинности высказываний:

- **а)** "наличия аттестата о среднем образовании достаточно для поступления в институт";
- **б)** "наличие аттестата о среднем образовании необходимо для поступления в институт";
- **в)** "если целое число делится на 6, то оно делится на 3";
- **г)** "подобие треугольников является необходимым условием их равенства";
- **д)** "подобие треугольников является необходимым и достаточным условием их равенства";
- **е)** "треугольники подобны только в случае их равенства";
- **ж)** "треугольники равны только в случае их подобия";
- **з)** "равенство треугольников является достаточным условием их подобия";
- **и)** "для того, чтобы треугольники были неравны, достаточно, чтобы они были неподобны";
- **к)** "для того, чтобы четырёхугольник был квадратом, достаточно, чтобы его диагонали были равны и перпендикулярны".

5.7. Подставьте в приведённые ниже высказывательные формы вместо логических переменных a , b , c , d такие высказывания, чтобы полученные таким образом составные высказывания имели смысл в повседневной жизни:

- **а)** если (a или (b и c)), то d ;
- **б)** если ($не a$ и $не b$), то (c или d);
- **в)** (a или b) тогда и только тогда, когда (c и $не d$).

5.8. Формализуйте следующий вывод: "Если a и b истинны, то c — истинно. Но c — ложно: значит, a или b ложны".

5.9. Формализуйте предостережение, которое одна жительница древних Афин сделала своему сыну, собиравшемуся заняться политической деятельностью: "Если ты будешь говорить правду, то тебя возненавидят люди. Если ты будешь лгать, то тебя возненавидят боги. Но ты должен говорить правду или лгать. Значит, тебя возненавидят люди или возненавидят боги".

Формализуйте также ответ сына: "Если я буду говорить правду, то боги будут любить меня. Если я буду лгать, то люди будут любить меня. Но я должен говорить правду или лгать. Значит, меня будут любить боги или меня будут любить люди".

5.10. Пусть a = "это утро ясное", a b = "это утро теплое". Выразите следующие формулы на обычном языке:

$$\begin{array}{llll}
 \text{а)} & a \cdot b & \text{з)} & \overline{a \vee b} & \text{ж)} & \overline{a \cdot b} & \text{к)} & \overline{a \rightarrow b} \\
 \text{б)} & a \cdot \overline{b} & \text{д)} & a \vee \overline{b} & \text{з)} & \overline{a \vee b} & \text{л)} & \overline{a \rightarrow b} \\
 \text{в)} & \overline{a \cdot b} & \text{е)} & \overline{a \vee b} & \text{и)} & \overline{a \cdot b} & \text{м)} & \overline{a \rightarrow b}
 \end{array}$$

5.11. Из двух данных высказываний a и b постройте составное высказывание, которое было бы:

- а) истинно тогда и только тогда, когда оба данных высказывания ложны;
- б) ложно тогда и только тогда, когда оба данных высказывания истинны.

5.12. Из трех данных высказываний a , b , c постройте составное высказывание, которое истинно, когда истинно какое-либо одно из данных высказываний, и только в этом случае.

Ответ: $a \cdot \bar{b} \cdot \bar{c} \vee \bar{a} \cdot b \cdot \bar{c} \vee \bar{a} \cdot \bar{b} \cdot c$.

5.13. Определите с помощью таблиц истинности, какие из следующих формул являются тождественно истинными или тождественно ложными:

а) $\bar{a} \cdot \bar{a} \vee b \cdot (a \cdot b \vee b)$

д) $a \cdot (b \cdot (\bar{a} \vee \bar{b}))$

б) $((a \vee \bar{b}) \rightarrow b) \cdot (\bar{a} \vee b)$

е) $\overline{(\bar{a} \vee \bar{b}) \cdot (\bar{b} \vee c) \vee \bar{a} \vee c}$

в) $a \cdot b \leftrightarrow (\bar{a} \vee \bar{b})$

ж) $\overline{(a \rightarrow b) \leftrightarrow (b \rightarrow a)}$

г) $a \cdot b \cdot (c \vee \bar{c} \vee d) \cdot \bar{b}$

5.14. Упростите следующие формулы, используя законы склеивания:

- а) $a \cdot b \cdot c \vee \bar{a} \cdot b \cdot c$
- б) $a \cdot b \cdot c \vee a \cdot \bar{b} \cdot c$
- в) $a \cdot b \cdot c \vee a \cdot \bar{b} \cdot c$
- г) $(a \vee b \vee c) \cdot (a \vee \bar{b} \vee c)$
- д) $(\bar{a} \vee \bar{b} \vee \bar{c}) \cdot (\bar{a} \vee b \vee c)$

Решение: $(\bar{a} \vee \bar{b} \vee \bar{c}) \cdot (\bar{a} \vee b \vee c) = (\bar{a} \vee \bar{b} \cdot c) \cdot (\bar{a} \vee b \cdot c) = \bar{a}$.

5.15. Упростите следующие формулы, используя законы поглощения:

- а) $a \vee a \cdot b \vee a \cdot b \cdot c \vee a \cdot d \cdot f$
- б) $a \cdot b \vee a \cdot b \cdot c \vee a \cdot b \cdot d$
- в) $a \cdot (a \vee b) \cdot (a \vee c)$
- г) $a \cdot b \cdot (a \cdot c \vee a \cdot b)$

5.16. Постройте таблицы истинности для логических формул и упростите формулы, используя законы алгебры логики:

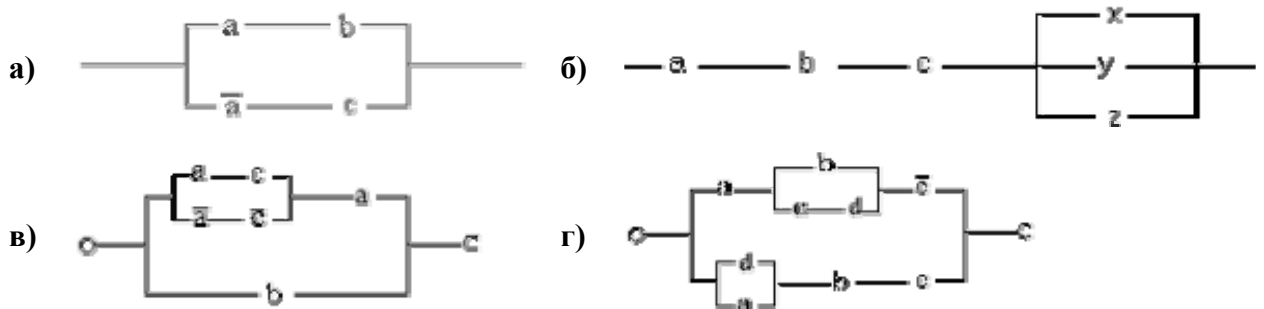
- а) $a \cdot \bar{c} \vee c \cdot (b \vee \bar{c}) \vee (a \vee \bar{b}) \cdot c$
- б) $a \cdot (b \vee \bar{c}) \vee \bar{a} \cdot b$
- в) $(\bar{a} \vee c) \cdot \overline{a \cdot c} \cdot (b \vee \bar{c}) \cdot \bar{b} \cdot c$
- г) $a \cdot b \cdot c \vee a \cdot \bar{b} \cdot c \vee a \cdot b \cdot \bar{c} \cdot d$
- е) $a \vee b \vee \bar{b} \cdot c \cdot d \vee \bar{b} \cdot \bar{c} \cdot \bar{d} \vee \bar{b} \cdot \bar{c} \cdot d$
- ж) $a \vee d \vee \bar{a} \cdot b \cdot c \vee \bar{a} \cdot \bar{b} \cdot c \vee \bar{a} \cdot \bar{b} \cdot \bar{c}$

- з) $\overline{a \vee b \vee c \vee \bar{b} \vee (a \vee \bar{b} \vee c \cdot \bar{a} \vee b \vee c) \vee \bar{a} \cdot \bar{b}}$
- и) $a \cdot b \cdot c \vee a \cdot b \cdot \bar{c} \vee a \cdot b \cdot c \cdot d \vee a \cdot b \cdot c \cdot \bar{d} \vee a \cdot b \cdot c \cdot d$
- к) $a \cdot d \cdot (\bar{a} \vee \bar{c} \cdot b \vee d) \vee a \cdot \bar{c} \vee \bar{a} \cdot b \cdot \bar{c}$

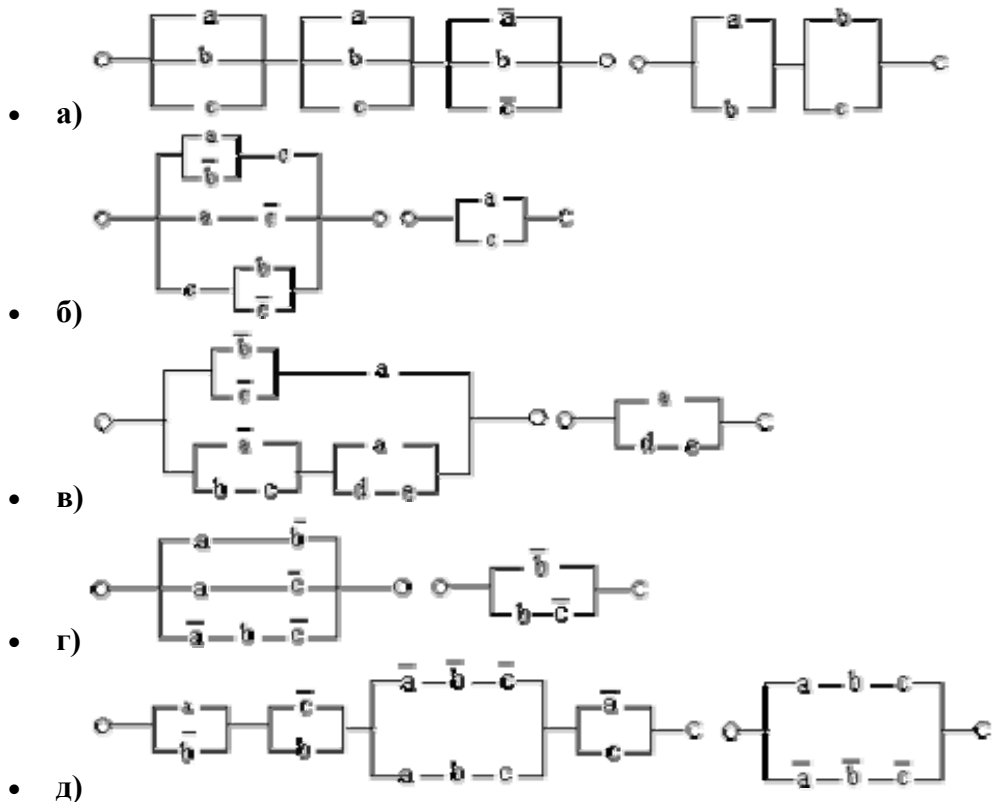
5.17. Приведите примеры переключательных схем, содержащих хотя бы два переключателя, функция проводимости которых

- а) тождественно равна единице;
- б) тождественно равна нулю.

5.18. Найдите функции проводимости следующих переключательных схем:



5.19. Проверьте равносильность следующих переключательных схем:



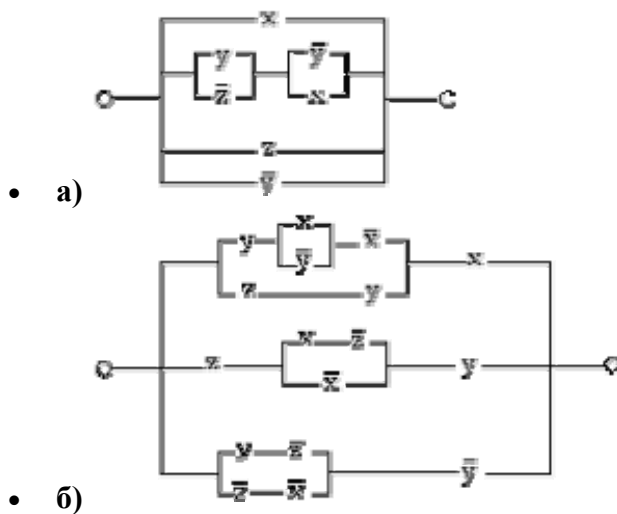
5.20. Постройте переключательные схемы с заданными функциями проводимости:

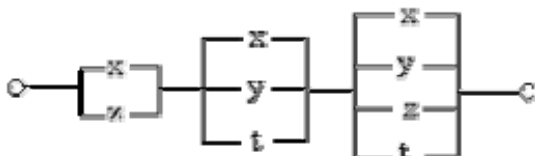
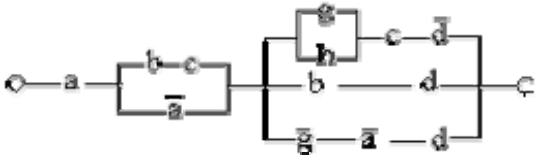
- а) $(a \vee b) \cdot c \vee \bar{a} \cdot b$
- б) $a \cdot (\bar{b} \vee c) \vee b \cdot (\bar{a} \vee c)$
- в) $a \vee \bar{b} \cdot c \cdot d \vee \bar{b} \cdot \bar{c} \cdot d \vee b \vee \bar{b} \cdot \bar{c} \cdot \bar{d}$
- г) $a \cdot b \cdot (c \cdot \bar{d} \vee \bar{b} \vee a) \vee \bar{a} \cdot (c \vee d) \cdot (a \vee b)$

5.21. Упростите функции проводимости и постройте переключательные схемы, соответствующие упрощенным функциям:

- а) $F = (\bar{a} \vee b) \cdot (a \vee e) \cdot (b \vee e)$
- б) $F = a \cdot b \cdot c \cdot c \vee b \cdot c \cdot d \vee b \cdot f$
- в) $F = a \cdot (\bar{b} \vee c \cdot (\bar{d} \cdot c \vee \bar{g}))$
- г) $F = a \cdot (\bar{b} \vee c) \vee a \cdot b \vee a \cdot e$
- д) $F = \overline{\bar{a} \cdot b \vee a \cdot (b \vee \bar{e})}$
- е) $F = (a \vee b \vee c) \cdot (\bar{a} \vee b \vee c) \cdot (a \vee \bar{b} \vee c)$
- ж) $F = a \cdot (\bar{b} \vee \bar{c}) \vee (a \vee c \cdot d) \cdot (\bar{a} \vee b \cdot e)$
- з) $F = a \cdot b \cdot c \vee a \cdot \bar{b} \cdot c \vee a \cdot b \cdot \bar{c} \cdot d$
- и) $F = a \vee \bar{a} \cdot (b \vee c) \vee (\bar{a} \vee d \vee g) \cdot (b \vee d) \cdot (c \vee \bar{d} \vee g \cdot h)$

5.22. Упростите следующие переключательные схемы:



- в) 
- г) 

ЛОГИЧЕСКИЕ ЗАДАЧИ

5.23. Три девочки — Роза, Маргарита и Анюта представили на конкурс цветоводов корзины выращенных ими роз, маргариток и анютиных глазок. Девочка, вырастившая маргаритки, обратила внимание Розы на то, что ни у одной из девочек имя не совпадает с названием любимых цветов. Какие цветы вырастила каждая из девочек?

5.24. Виновник ночного дорожно-транспортного происшествия скрылся с места аварии. Первый из опрошенных свидетелей сказал работникам ГАИ, что это были "Жигули", первая цифра номера машины — единица. Второй свидетель сказал, что машина была марки "Москвич", а номер начинался с семёрки. Третий свидетель заявил, что машина была иностранная, номер начинался не с единицы. При дальнейшем расследовании выяснилось, что каждый из свидетелей правильно указал либо только марку машины, либо только первую цифру номера. Какой марки была машина и с какой цифры начинался номер?

5.25. Пятеро одноклассников: Ирена, Тимур, Камилла, Эльдар и Залим стали победителями олимпиад школьников по физике, математике, информатике, литературе и географии. Известно, что:

- победитель олимпиады по информатике учит Ирену и Тимура работе на компьютере;
- Камилла и Эльдар тоже заинтересовались информатикой;
- Тимур всегда побаивался физики;
- Камилла, Тимур и победитель олимпиады по литературе занимаются плаванием;
- Тимур и Камилла поздравили победителя олимпиады по математике;
- Ирена сожалеет о том, что у нее остается мало времени на литературу.

Победителем какой олимпиады стал каждый из этих ребят?

5.26. Ирена любит мороженое с фруктами. В кафе был выбор из таких вариантов:

- пломбир с орехами;
- пломбир с бананами;
- пломбир с черникой;
- шоколадное с черникой;
- шоколадное с клубникой.

В четырёх вариантах Ирене не нравились или тип мороженого, или наполнитель, а в одном варианте ей не нравились ни мороженое, ни наполнитель. Она попросила приготовить из имеющихся продуктов порцию по своему вкусу. Какое же мороженое и с какими фруктами любит Ирена?

5.27. На очередном этапе автогонок "Формула 1" первые четыре места заняли Шумахер, Алеззи, Хилл и Кулхардт. Опоздавший к месту награждения телерепортёр успел заснять пилотов, занявших второе и третье места, которые поливали друг друга шампанским. В это время Шумахер с четвёртым гонщиком пожимали друг другу руки. Далее в кадр попал мокрый Хилл, поздравляющий пилота, занявшего второе место. Напоследок оператор снял сцену, в которой Шумахер и Кулхардт пытались втащить на пьедестал почёта пилота, занявшего четвёртое место. Просматривая отснятый материал, режиссёр спортивного выпуска быстро разобрался, кто из пилотов какое место занял. Он знал, что, в соответствии с церемонией награждения победителей гонок, пилоты, занявшие первые три места, поливают друг друга шампанским из огромных бутылок знаменитой фирмы — спонсора соревнований. Какое же место занял каждый пилот?

5.28. В некотором царстве-государстве повадился Змей Горыныч разбойничать. Послал царь четырёх богатырей погубить Змея, а награду за то обещал великую. Вернулись богатыри с победой и спрашивает их царь: "Так кто же из вас главный победитель, кому достанется царёва дочь и полцарства?" Засмутились добры молодцы и ответы дали туманные: Сказал Илья Муромец: "Это все Алеша Попович, царь-батюшка". Алеша Попович возразил: "То был Микула Селянинович". Микула Селянинович: "Не прав Алеша, не я это". Добрыня Никитич: "И не я, батюшка". Подвернулась тут баба Яга и говорит царю: "А прав то лишь один из богатырей, видела я всю битву своими глазами". Кто же из богатырей победил Змея Горыныча?

5.29. При составлении расписания на пятницу были высказаны пожелания, чтобы информатика была первым или вторым уроком, физика — первым или третьим, история — вторым или третьим. Можно ли удовлетворить одновременно все высказанные пожелания?

5.30. Обсуждая конструкцию нового трёхмоторного самолёта, трое конструкторов поочередно высказали следующие предположения:

- 1) при отказе второго двигателя надо приземляться, а при отказе третьего можно продолжать полёт;
- 2) при отказе первого двигателя лететь можно, или при отказе третьего двигателя лететь нельзя;
- 3) при отказе третьего двигателя лететь можно, но при отказе хотя бы одного из остальных надо садиться.

Лётные испытания подтвердили правоту каждого из конструкторов. Определите, при отказе какого из двигателей нельзя продолжать полёт.

5.31. В соревнованиях по плаванию участвовали Андрей, Виктор, Саша и Дима. Их друзья высказали предположения о возможных победителях:

- 1) первым будет Саша, Виктор будет вторым;

- 2) вторым будет Саша, Дима будет третьим;
3) Андрей будет вторым, Дима будет четвёртым.

По окончании соревнований оказалось, что в каждом из предположений только одно из высказываний истинно, другое ложно.

Какое место на соревнованиях занял каждый из юношей, если все они заняли разные места.

5.32. Для длительной международной экспедиции на околоземной космической станции надо из восьми претендентов отобрать шесть специалистов: по авиации, космонавигации, биомеханике, энергетике, медицине и астрофизике. Условия полёта не позволяют совмещать работы по разным специальностям, хотя некоторые претенденты владеют двумя специальностями. Обязанности авианта могут выполнять Геррети и Нам; космонавигатора — Кларк и Фриш; биомеханика — Фриш и Нам; энергетика — Депардьё и Леонов; врача — Депардьё и Хорхес; астрофизика — Волков и Леонов.

По особенностям психологической совместимости врачи рекомендуют совместные полеты Фриша и Кларка, а также Леонова с Хорхесом и Депардьё. Напротив, нежелательно, чтобы Депардьё оказался в одной экспедиции с Намом, а Волков — с Кларком.

Кого следует включить в состав экспедиции?

Ответы — Раздел 5. Логические основы компьютеров

5.1. Являются высказываниями: а), г), д), ж), з), и), к);
не являются высказываниями: б); в); е).

5.2. Истинные: д), з), к);
ложные: а), и);
истинность трудно установить: г);
можно рассматривать и как истинное, и как ложное в зависимости от требуемой точности представления: ж).

5.3. Образцы.
Истинные высказывания: а) “ $2+2=4$ ”; б) “сила притяжения тел обратно пропорциональна квадрату расстояния между ними”; в) “зайцы питаются растениями”; г) “бит - фундаментальная единица информации, используемая в теории информации”; д) “два треугольника равны, если две стороны и угол между ними одного треугольника равны двум сторонам и углу между ними другого треугольника”; е) “понедельник - первый день недели”.

Ложные высказывания: а) “ $4+3=5$ ”; б) “тело падает на Землю с ускорением, пропорциональным своей массе”; в) “животные это неживая природа”; г) “информатика - наука о термической обработке металлов”; д) “квадрат это фигура у которой пять сторон”; е) “лев - домашнее животное”.

5.4. а) “Эльбрус – не высочайшая горная вершина Европы”; б) “ $2 < 5$ ”; в) “ $10 >= 7$ ”; г) “не все натуральные числа целые”; д) “не через любые три точки на плоскости можно провести окружность”; е) “теннисист Кафельников проиграл финальную игру”; ж) “мишень не поражена первым выстрелом”; з) “это утро не ясное или оно не теплое” (Пояснение. Пусть $A =$ “это утро ясное”, а $B =$ “это утро теплое”. Тогда “это утро ясное и теплое” можно записать как $A \cdot B$, отрицанием чего является $\overline{A \cdot B} = \overline{A} \vee \overline{B}$, что соответствует высказывательной форме “это утро не ясное или оно не не теплое”;

и) “число n не делится на 2 и оно делится на 3”; к) “этот треугольник не равнобедренный или он не прямоугольный”; л) “не каждый ученик писал контрольную своей ручкой” (вариант: “кто-то писал контрольную не своей ручкой”).

5.5. Являются отрицаниями друг друга: б), г), д), к);
не являются отрицаниями друг друга: а), в), е), ж), з), и).

5.6. Истинны: б), в), г), з), к), и);
ложны: а), д), е), ж).

5.8. $((a \cdot b \rightarrow c) \cdot \bar{c}) \rightarrow (\bar{a} \vee \bar{b})$.

5.9. Решение. Введем обозначения для логических высказываний: а – “ты будешь говорить правду”; б – “тебя возненавидят люди”; с – “тебя возненавидят боги”. Договоримся считать, что некоторое заданное высказывание x истинно, если нет оговорки. Тогда предостережение матери можно записать так:

$((a \rightarrow b) \vee (a \rightarrow c)) \cdot (a \rightarrow \bar{a}) \rightarrow (b \vee c)$. А ответ сына – так:

$((a \rightarrow \bar{c}) \vee (a \rightarrow \bar{b})) \cdot (a \rightarrow \bar{a}) \rightarrow (\bar{b} \vee \bar{c})$.

5.10.

- | | |
|---|---|
| а) “это утро ясное и тёплое”; | ж) “это утро не ясное или не тёплое”; |
| б) “это утро ясное и оно не тёплое”; | з) “это утро не ясное и не тёплое”; |
| в) “это утро не ясное и оно не тёплое”; | и) “это утро ясное или не тёплое”; |
| г) “это утро не ясное или оно не тёплое”; | к) “если это утро ясное, то оно не тёплое”; |
| д) “это утро ясное или оно не тёплое”; | л) “если это утро не ясное, то оно тёплое”; |
| е) “это утро не ясное или оно не тёплое”; | м) “это утро ясное и не тёплое”. |
-

5.11. а) $\bar{a} \cdot \bar{b}$; б) $\overline{a \cdot b}$.

5.13. Тождественно истинные: а), в), е);
тождественно ложные: г), д), ж).

5.14. а) $b \cdot c$; б) а; в) $c \cdot (a \vee b) \vee a \cdot b$ (Указание: повторить четвертое логическое слагаемое 3 раза); г) $a \vee c$.

5.15. а) а; б) $a \cdot b$; в) а; г) $a \cdot b$;

5.16. а) $a \vee c$; б) $\bar{b} \cdot (\bar{a} \vee c)$; в) $\overline{a \vee c}$; г) $a \vee c$; д) $a \cdot (c \vee b \cdot d)$; е) $a \vee b \vee \bar{c} \vee d$; ж) $a \vee \bar{b} \vee c \vee d$; з) \bar{b} ; и) $a \cdot (b \vee c \cdot d)$; к) $a \cdot d \vee c \cdot (a \vee b)$.

а) $F = a \cdot b \vee \bar{a} \cdot c$ б) $F = a \cdot b \cdot c \cdot (x \vee y \vee z)$
5.18. в) $F = b \vee (a \cdot c \vee \bar{a} \cdot \bar{c}) \cdot a$ г) $F = a \cdot (b \vee c \cdot d) \cdot \bar{c} \vee (a \vee d) \cdot b \cdot c$

5.19. Равносильны: б), в), д);
неравносильны: а), г).
Пояснения. Обозначим функции проводимости рассматриваемых переключательных схем как F_1 и F_2 , соответственно. Тогда: а) $F_1 = (a \vee b \vee c) \cdot (\bar{a} \vee b \vee c) \cdot (\bar{a} \vee b \vee \bar{c}) = b \vee a \cdot \bar{c}$;
 $F_2 = (a \vee b) \cdot (b \vee c) = b \vee a \cdot c$, $F_1 \neq F_2$;
б) $F_1 = (a \vee \bar{b}) \cdot c \vee a \cdot \bar{c} \vee c \cdot (b \vee \bar{c}) = a \vee c$; $F_2 = a \vee c$, $F_1 = F_2$;
в) $F_1 = (\bar{b} \vee \bar{c}) \cdot a \vee (\bar{a} \vee b \cdot c) \cdot (a \vee d \cdot e) = a \vee d \cdot e$; $F_2 = a \vee d \cdot e$, $F_1 = F_2$;
д) $F_1 = (a \vee \bar{b}) \cdot (b \vee \bar{c}) \cdot (\bar{a} \cdot \bar{b} \cdot \bar{c} \vee a \cdot b \cdot c) \cdot (\bar{a} \vee c) = a \cdot b \cdot c \vee \bar{a} \cdot \bar{b} \cdot \bar{c}$; $F_2 = a \cdot b \cdot c \vee \bar{a} \cdot \bar{b} \cdot \bar{c}$, $F_1 = F_2$.

5.21. Упрощенные функции:
а) $\bar{a} \cdot c \vee b \cdot (a \vee c)$; б) $b \cdot (c \cdot (a \cdot e \vee d) \vee f)$ в) $\bar{a} \vee b \cdot (\bar{c} \vee (d \vee \bar{e}) \cdot g)$;
г) а; д) $\bar{b} \cdot (\bar{a} \vee c)$; е) $c \vee a \cdot b$;
ж) $a \vee e \cdot d$; з) $a \cdot (c \vee b \cdot d)$; и) $a \vee b \vee c \vee d \cdot g \cdot h$;

5.22. Функции проводимости упрощенных схем:

а) $F = x \vee y \vee z$, б) $F = y \cdot z \vee x \cdot y \cdot z$, в) $F = z$, г) $F = x \vee z \cdot (y \vee t)$, д) $F = a \cdot b \cdot c \cdot (g \vee h \vee d)$.

5.23. Аня вырастила маргаритки, Роза – анютины глазки, Маргарита – розы.

5.24. “Жигули”, номер начинается с семерки.

5.25. Ирена – победитель олимпиады по математике, Тимур – по географии, Камилла – по физике, Эльдар – по литературе, Залим – по информатике.

5.26. Пломбир с клубникой. Пояснение. Два других возможных варианта – шоколадное с орехами и шоколадное с бананами, не подходят по условию задачи (число вариантов, в которых не нравятся и тип мороженого и наполнитель, в этих случаях равно двум вместо одного).

5.27. Шумахер пришел первым, Кулхардт – вторым, Хилл – третьим и Алези – четвертым. Пояснение. Шумахер не четвертый, так как он пожимал четвертому руку; он не второй и не третий, так как пилоты, занявшие эти места поливали друг друга шампанским в то время как он пожимал руку. Следовательно, Шумахер первый. Далее, раз Хилл мокрый, то он занял одно из призовых мест, но не первое и не второе (поздравлял пилота, занявшего второе место). Следовательно, Хилл – третий. Кулхардт занял не четвертое место, так как он втаскивал на пьедестал пилота, занявшего четвертое место. Следовательно, он второй.

5.28. Добрыня Никитич.

5.29. Есть две возможности:

- а) первый урок – информатика, второй – история, третий – физика;
б) первый урок – физика, второй – информатика, третий – история.
-

5.30. При отказе второго двигателя нельзя продолжать полет.

5.31. Первое место занял Саша, второе – Андрей, третье – Дима, четвертое – Виктор.

5.32. В экспедицию следует включить: аэронавтом – Геррети, космонавигатором – Кларка, биомехаником – Фриша, энергетиком – Депардье, врачом – Хорхеса, астрофизиком – Леонова.

6.1. Что такое программное обеспечение?

Под программным обеспечением (Software) понимается совокупность программ, выполняемых вычислительной системой.

К программному обеспечению (ПО) относится также вся область деятельности по проектированию и разработке ПО:

- технология проектирования программ (например, нисходящее проектирование, структурное и объектно-ориентированное проектирование и др.);
- методы тестирования программ [[ссылка](#), [ссылка](#)];
- методы доказательства правильности программ;
- анализ качества работы программ;
- документирование программ;
- разработка и использование программных средств, облегчающих процесс проектирования программного обеспечения, и многое другое.

Программное обеспечение — неотъемлемая часть компьютерной системы. Оно является логическим продолжением технических средств. Сфера применения конкретного компьютера определяется созданным для него ПО.

Сам по себе компьютер не обладает знаниями ни в одной области применения.
Все эти знания сосредоточены в выполняемых на компьютерах программах.

Программное обеспечение современных компьютеров включает миллионы программ — от игровых до научных.

6.2. Как классифицируется программное обеспечение?

В первом приближении все программы, работающие на компьютере, можно условно разделить на три категории (рис. 6.1):

1. прикладные программы, непосредственно обеспечивающие выполнение необходимых пользователям работ;
2. системные программы, выполняющие различные вспомогательные функции, например:
 - управление ресурсами компьютера;
 - создание копий используемой информации;
 - проверка работоспособности устройств компьютера;
 - выдача справочной информации о компьютере и др.;
3. инструментальные программные системы, облегчающие процесс создания новых программ для компьютера.



Рис. 6.1. Категории программного обеспечения

При построении классификации ПО нужно учитывать тот факт, что стремительное развитие вычислительной техники и расширение сферы приложения компьютеров резко ускорили процесс эволюции программного обеспечения.

Если раньше можно было по пальцам перечислить основные категории ПО — операционные системы, трансляторы, пакеты прикладных программ, то сейчас ситуация коренным образом изменилась.

Развитие ПО пошло как вглубь (появились новые подходы к построению операционных систем, языков программирования и т.д.), так и вширь (прикладные программы перестали быть прикладными и приобрели самостоятельную ценность).

Соотношение между требующимися программными продуктами и имеющимися на рынке меняется очень быстро. Даже классические программные продукты, такие, как операционные системы, непрерывно развиваются и наделяются интеллектуальными функциями, многие из которых ранее относились только к интеллектуальным возможностям человека.

Кроме того, появились нетрадиционные программы, классифицировать которые по устоявшимся критериям очень трудно, а то и просто невозможно, как, например, программа — *электронный собеседник*.

На сегодняшний день можно сказать, что **более или менее определённо сложились следующие группы программного обеспечения:**

- [операционные системы](#) и [оболочки](#);
- [системы программирования](#) (трансляторы, библиотеки подпрограмм, отладчики и т.д.);
- инструментальные системы;
- [интегрированные пакеты программ](#);
- динамические [электронные таблицы](#);
- системы машинной графики;
- системы управления базами данных ([СУБД](#));
- прикладное программное обеспечение.

Структура программного обеспечения показана на рис. 6.2. Разумеется, эту классификацию нельзя считать исчерпывающей, но она более или менее наглядно отражает направления совершенствования и развития программного обеспечения.

6.3. Какие программы называют прикладными?

Прикладная программа — это любая конкретная программа, способствующая решению какой-либо задачи в пределах данной проблемной области.

Например, там, где на компьютер возложена задача контроля за финансовой деятельностью какой-либо фирмы, прикладной будет программа подготовки платежных ведомостей.

Прикладные программы могут носить и общий характер, например, обеспечивать составление и печатание документов и т.п.

В противоположность этому, операционная система или инструментальное ПО не вносят прямого вклада в удовлетворение конечных потребностей пользователя.

Прикладные программы могут использоваться либо автономно, то есть решать поставленную задачу без помощи других программ, либо в составе программных комплексов или пакетов.

6.4. Какова роль и назначение системных программ?

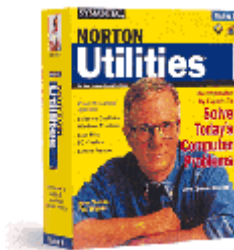
Системные программы выполняются вместе с прикладными и служат для управления ресурсами компьютера — центральным процессором, памятью, вводом-выводом.

Это программы общего пользования, которые **предназначены для всех пользователей компьютера**. Системное программное обеспечение разрабатывается так, чтобы компьютер мог эффективно выполнять прикладные программы.



Рис. 6.2. Структура программного обеспечения компьютера

Среди десятков тысяч системных программ особое место занимают **операционные системы**, которые обеспечивают управление **ресурсами компьютера** с целью их эффективного использования.



Важными классами системных программ являются также программы вспомогательного назначения — **утилиты** (лат. *utilitas* — польза). Они либо **расширяют и дополняют соответствующие возможности операционной системы**, либо **решают самостоятельные важные задачи**. Кратко опишем некоторые разновидности утилит:

- **программы контроля, тестирования и диагностики**, которые используются для проверки правильности функционирования устройств компьютера и для обнаружения неисправностей в процессе эксплуатации; указывают причину и место неисправности;
- **программы-драйверы**, которые расширяют возможности операционной системы по управлению устройствами ввода-вывода, оперативной памятью и т.д.; с помощью драйверов возможно подключение к компьютеру новых устройств или нестандартное использование имеющихся;
- **программы-упаковщики** (архиваторы), которые позволяют записывать информацию на дисках более плотно, а также объединять копии нескольких файлов в один архивный файл;
- **антивирусные программы**, предназначенные для предотвращения заражения компьютерными вирусами и ликвидации последствий заражения вирусами;

Компьютерный вирус — это специально написанная небольшая по размерам программа, которая может "приписывать" себя к другим программам для выполнения каких-либо вредных действий — портит файлы, "засоряет" оперативную память и т.д.

- **программы оптимизации и контроля качества дискового пространства** ;
- **программы восстановления информации, форматирования, защиты данных** ;
- **коммуникационные программы**, организующие обмен информацией между компьютерами;
- **программы для управления памятью**, обеспечивающие более гибкое использование оперативной памяти;
- **программы для записи CD-ROM, CD-R** и многие другие.

Часть утилит входит в состав операционной системы, а другая часть функционирует независимо от нее, т.е. автономно.

6.5. Что такое операционная система?

Операционная система — это комплекс взаимосвязанных системных программ, назначение которого — организовать взаимодействие пользователя с компьютером и выполнение всех других программ.

Операционная система выполняет роль связующего звена между аппаратурой компьютера, с одной стороны, и выполняемыми программами, а также пользователем, с другой стороны.

Операционная система обычно хранится во внешней памяти компьютера — *на диске*. При включении компьютера она считывается с дисковой памяти и размещается в *ОЗУ*.

Этот процесс называется *загрузкой операционной системы*.

В функции операционной системы входит:

- осуществление диалога с пользователем;
- ввод-вывод и управление данными;
- планирование и организация процесса обработки программ;
- распределение ресурсов (оперативной памяти и кэша, процессора, внешних устройств);
- запуск программ на выполнение;
- всевозможные вспомогательные операции обслуживания;
- передача информации между различными внутренними устройствами;
- программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтера и др.).

Операционную систему можно назвать программным продолжением устройства управления компьютера. Операционная система скрывает от пользователя сложные ненужные подробности взаимодействия с аппаратурой, образуя прослойку между ними. В результате этого люди освобождаются от очень трудоёмкой работы по организации взаимодействия с аппаратурой компьютера.

В зависимости от количества одновременно обрабатываемых задач и числа пользователей, которых могут обслуживать ОС, различают четыре основных класса операционных систем:

1. **однопользовательские однозадачные**, которые поддерживают одну клавиатуру и могут работать только с одной (в данный момент) задачей;
2. **однопользовательские однозадачные с фоновой печатью**, которые позволяют помимо основной задачи запускать одну дополнительную задачу, ориентированную, как правило, на вывод информации на печать. Это ускоряет работу при выдаче больших объёмов информации на печать;
3. **однопользовательские многозадачные**, которые обеспечивают одному пользователю параллельную обработку нескольких задач. Например, к одному компьютеру можно подключить несколько принтеров, каждый из которых будет работать на "свою" задачу;
4. **многопользовательские многозадачные**, позволяющие на одном компьютере запускать несколько задач нескольким пользователям. Эти ОС очень сложны и требуют значительных машинных ресурсов.

В различных моделях компьютеров используют операционные системы с разной архитектурой и возможностями. Для их работы требуются разные ресурсы. Они предоставляют разную степень сервиса для программирования и работы с готовыми программами.

Операционная система для персонального компьютера, ориентированного на профессиональное применение, должна содержать следующие основные компоненты:

- программы управления вводом/выводом;
- программы, управляющие файловой системой и планирующие задания для компьютера;
- процессор командного языка, который принимает, анализирует и выполняет команды, адресованные операционной системе.

Каждая операционная система имеет свой командный язык, который позволяет пользователю выполнять те или иные действия:

- обращаться к каталогу;
- выполнять разметку внешних носителей;
- запускать программы;
- ... другие действия.

Анализ и исполнение команд пользователя, включая загрузку готовых программ из файлов в оперативную память и их запуск, осуществляет командный процессор операционной системы.

Для управления внешними устройствами компьютера используются специальные системные программы — драйверы. Драйверы стандартных устройств образуют в совокупности базовую систему ввода-вывода (BIOS), которая обычно заносится в постоянное ЗУ компьютера.

6.6. Что такое файловая система ОС?

Файл (англ. *file* — папка) — это именованная совокупность любых данных, размещенная на внешнем запоминающем устройстве и хранимая, пересылаемая и обрабатываемая как единое целое. Файл может содержать программу, числовые данные, текст, закодированное изображение и др.

Файловая система — это средство для организации хранения файлов на каком-либо носителе.

Файлы физически реализуются как *участки памяти на внешних носителях* — магнитных дисках или CD-ROM. Каждый файл занимает некоторое количество блоков дисковой памяти. Обычная длина блока — 512 байт.

Обслуживает файлы специальный модуль операционной системы, называемый драйвером файловой системы. Каждый файл имеет имя, зарегистрированное в каталоге — оглавлении файлов.

Каталог (иногда называется *директорией* или *папкой*) доступен пользователю через командный язык операционной системы. Его можно просматривать, переименовывать зарегистрированные в нем файлы, переносить их содержимое на новое место и удалять.

Каталог может иметь собственное имя и храниться в другом каталоге наряду с обычными файлами: так образуются иерархические файловые структуры. Пример такой структуры — на рис. 6.3.

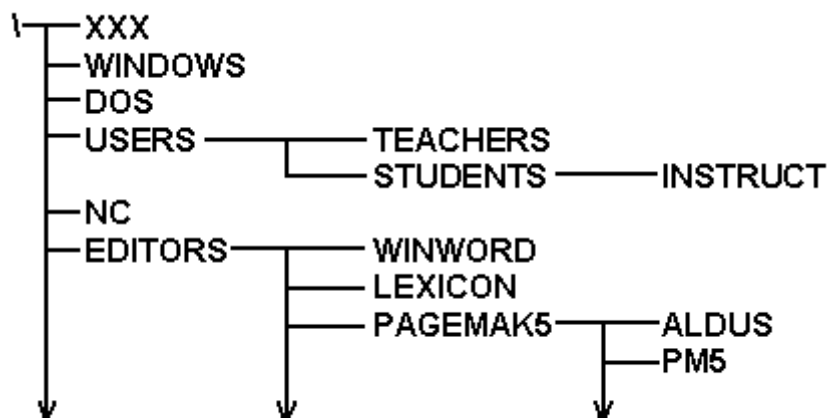


Рис. 6.3. Дерево каталогов на диске

Что происходит, когда пользователь подает операционной системе команду "*открыть файл ...*", в которой указано *имя файла* и *имя каталога*, в котором размещён этот файл?

Для выполнения этой команды драйвер файловой системы обратится к своему *справочнику*, выясняет, какие блоки диска соответствуют указанному файлу, а затем передает запрос на считывание этих блоков драйверу диска.

При выполнении команды "*сохранить файл*" драйвер файловой системы ищет на диске незанятые блоки, отмечает их, как распределённые для вновь созданного файла, и передаёт драйверу диска запрос на запись в эти блоки данных пользователя.

Драйвер файловой системы обеспечивает доступ к информации, записанной на магнитный диск, по имени файла и распределяет пространство на магнитном диске между файлами.

Для выполнения этих функций драйвер файловой системы хранит на диске не только информацию пользователя, но и свою собственную служебную информацию. **В служебных областях диска хранится список всех файлов и каталогов**, а также различные дополнительные справочные таблицы, служащие для повышения скорости работы драйвера файловой системы.

К файловой системе имеет доступ также и любая прикладная программа, для чего во всех языках программирования имеются специальные процедуры.

Понятие файла может быть обращено на любой источник или потребитель информации в машине, например, в качестве файла для программы могут выступать *принтер, дисплей, клавиатура* и др.

Структура файловой системы и структура хранения данных на внешних магнитных носителях определяет удобство работы пользователя, скорость доступа к файлам и т.д.

6.7. Какова структура операционной системы MS DOS?

Операционная система MS DOS (Microsoft Disk Operating System) — самая распространенная ОС на 16-разрядных персональных компьютерах. Она состоит из следующих основных модулей (рис. 6.4):

- базовая система ввода/вывода ([BIOS](#));
- блок начальной загрузки (Boot Record);
- модуль расширения базовой системы ввода/вывода (IO.SYS);
- модуль обработки прерываний (MSDOS.SYS);
- командный процессор (COMMAND.COM);
- [утилиты](#) MS DOS.

Каждый из указанных модулей выполняет определенную часть функций, возложенных на ОС. Места постоянного размещения этих модулей различны. Так, *базовая система ввода/вывода находится в постоянном запоминающем устройстве (ПЗУ)*, а не на дисках, как все остальные модули.

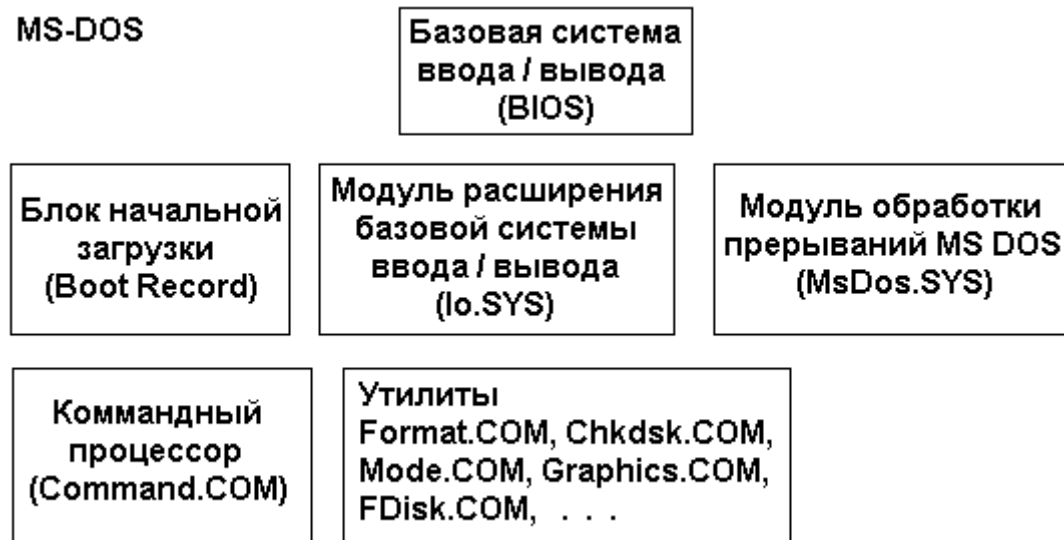


Рис. 6.4. Состав операционной системы MS-DOS

Базовая система ввода/вывода (BIOS) выполняет наиболее простые и универсальные услуги операционной системы, связанные с осуществлением *ввода-вывода*. В функции BIOS входит также *автоматическое тестирование основных аппаратных компонентов* (оперативной памяти и др.) при включении машины и *вызов блока начальной загрузки DOS*.

Блок начальной загрузки (или просто *загрузчик*) — это очень короткая программа, единственная функция которой заключается в считывании с диска в оперативную память двух других частей DOS — модуля расширения базовой системы ввода/вывода и модуля обработки прерываний.

Модуль расширения базовой системы ввода/вывода дает возможность использования *дополнительных драйверов*, обслуживающих новые внешние устройства, а

также драйверов для нестандартного обслуживания внешних устройств.

Модуль обработки прерываний реализует основные высокоуровневые услуги DOS, поэтому его и называют основным.

Командный процессор DOS обрабатывает команды, вводимые пользователем.

Утилиты DOS — это программы, поставляемые вместе с операционной системой в виде отдельных файлов. Они выполняют действия обслуживающего характера, например, разметку дискет, проверку дисков и т.д.

6.8. Что такое программы-оболочки?

Оболочки — это программы, созданные для упрощения работы со сложными программными системами, такими, например, как DOS. Они преобразуют неудобный командный пользовательский интерфейс в дружелюбный графический интерфейс или интерфейс типа "меню". Оболочки предоставляют пользователю удобный доступ к файлам и обширные сервисные услуги.

Самая популярная у пользователей IBM-совместимого ПК оболочка — пакет программ **Norton Commander**. Он обеспечивает:

- создание, копирование, пересылку, переименование, удаление, поиск файлов, а также изменение их атрибутов;
- отображение дерева каталогов и характеристик входящих в них файлов в форме, удобной для восприятия человека;
- создание, обновление и распаковку архивов (групп сжатых файлов);
- просмотр текстовых файлов;
- редактирование текстовых файлов;
- выполнение из её среды практически всех команд DOS;
- запуск программ;
- выдачу информации о ресурсах компьютера;
- создание и удаление каталогов;
- поддержку межкомпьютерной связи;
- поддержку электронной почты через модем.

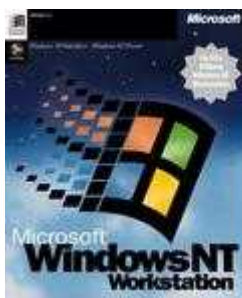
В начале 90-х годов во всем мире огромную популярность приобрела графическая оболочка **MS-Windows 3.x**, преимущество которой состоит в том, что она облегчает использование компьютера, и её графический интерфейс вместо набора сложных команд с клавиатуры позволяет выбирать их мышью из меню практически мгновенно. Операционная среда Windows, работающая совместно с операционной системой DOS, реализует все свойства, необходимые для производительной работы пользователя, в том числе — многозадачный режим.

Оболочка **Norton Navigator** — это набор мощных программ для управления файлами, расширяющий возможности Windows. Позволяет экономить время практически на всех операциях: поиск файлов, копирование и перемещение файлов, открытие каталогов.

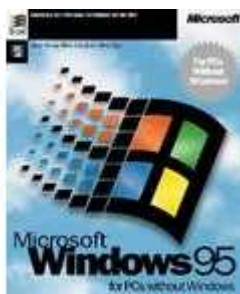
6.9. Что собой представляют операционные системы *Windows*, *Unix*, *Linux* ?

Операционные системы *Windows*

В настоящее время большинство компьютеров в мире работают под управлением той или иной версии операционной среды *Windows* фирмы *Microsoft*. Охарактеризуем наиболее распространенные версии.



Windows NT (*NT* — англ. *New Technology*) — это операционная система, а не просто графическая оболочка. Она использует все возможности новейших моделей персональных компьютеров и работает без *DOS*. *Windows NT* — 32-разрядная ОС со встроенной сетевой поддержкой и развитыми многопользовательскими средствами. Она предоставляет пользователям истинную многозадачность, многопроцессорную поддержку, секретность, защиту данных и многое другое. Эта операционная система очень удобна для пользователей, работающих в рамках локальной сети, для коллективных пользователей, особенно для групп, работающих над большими проектами и обменивающимися данными.



Windows 95 представляет собой универсальную высокопроизводительную многозадачную и многопоточную 32-разрядную ОС нового поколения с графическим интерфейсом и расширенными сетевыми возможностями. *Windows 95* — интегрированная среда, обеспечивающая эффективный обмен информацией между отдельными программами и предоставляющая пользователю широкие возможности работы с мультимедиа, обработки текстовой, графической, звуковой и видеoinформации. Интегрированность подразумевает также *совместное использование ресурсов компьютера всеми программами*.

Эта операционная система *обеспечивает работу пользователя в сети*, предоставляя встроенные средства поддержки для обмена файлами и меры по их защите, возможность совместного использования принтеров, факсов и других общих ресурсов. *Windows 95* позволяет отправлять сообщения электронной почтой, факсимильной связью, поддерживает удаленный доступ. Применяемый в *Windows 95* защищенный режим не позволяет прикладной программе в случае сбоя нарушить работоспособность системы,

надежно предохраняет приложения от случайного вмешательства одного процесса в другой, обеспечивает определенную устойчивость к вирусам.

Пользовательский интерфейс *Windows 95* прост и удобен. В отличие от оболочки *Windows 3* эта операционная система не нуждается в установке на компьютере операционной системы DOS. Она предназначена для установки на настольных ПК и компьютерах блокнотного типа с процессором *486* или *Pentium*. Рекомендуемый размер оперативной памяти 32—128 Мбайт. После включения компьютера и выполнения тестовых программ BIOS операционная система *Windows 95* автоматически загружается с жесткого диска. После загрузки и инициализации системы на экране появляется *рабочий стол*, на котором размещены различные *графические объекты*. Пользовательский интерфейс спроектирован так, чтобы максимально облегчить усвоение этой операционной системы новичками и создать комфортные условия для пользователя.

Windows 98 отличается от *Windows 95* тем, что в ней операционная система объединена с браузером ***Internet Explorer*** посредством интерфейса, выполненного в виде *Web*-браузера и оснащенного кнопками "Назад" и "Вперед" для перехода на предыдущую и последующую *Web*-страницы. Кроме этого, в ней улучшена совместимость с новыми аппаратными средствами компьютера, она одинаково удобна как для использования на настольных, так и на портативных компьютерах.

Windows 2000 Professional — операционная система нового поколения для делового использования на самых разнообразных компьютерах — от портативных до серверов. Эта ОС является наилучшей для ведения коммерческой деятельности в Интернете. Она объединяет присущую *Windows 98* простоту использования в Интернете, на работе, в пути с присущими *Windows NT* надежностью, экономичностью и безопасностью.

Windows CE 3.0 — операционная система для **мобильных вычислительных устройств**, таких, как карманные компьютеры, цифровые информационные пейджеры, сотовые телефоны, мультимедийные и развлекательные приставки, включая DVD проигрыватели и устройства целевого доступа в Интернет.

Операционная система *Windows CE* — 32-разрядная, многозадачная, многопоточная операционная система, имеющая открытую архитектуру, разрешающую использование множеств устройств. *Windows CE* позволяет устройствам различных категорий "говорить" и обмениваться информацией друг с другом, связываться с корпоративными сетями и с Интернет, пользоваться электронной почтой.

Windows CE компактна, но высоко производительна. Это мобильная система, функционирующая с микропроцессорами различных марок и изготовителей. Для нее есть программы *Word* и *Excel*, которые совместимы с их настольными аналогами. Имеет интегрированную систему управления питанием.

Операционная система *Unix*

Операционная система *Unix* была создана в *Bell Telephone Laboratories*. С краткой историей ее создания можно ознакомиться [здесь](#). *Unix* — **многозадачная операционная система, способная обеспечить одновременную работу очень большого количество пользователей**. Ядро ОС *Unix* написано на языке высокого уровня *C* и имеет только около 10 процентов кода на ассемблере. Это позволяет за считанные месяцы переносить ОС *Unix* на другие аппаратные платформы и достаточно легко вносить в нее серьезные изменения и дополнения. ***UNIX* является первой действительно переносимой операционной системой. В многочисленные существующие версии *UNIX* постоянно вносятся изменения.** С одной стороны, это расширяет возможности системы, делает ее

мощнее и надежнее, с другой — ведет к появлению различий между существующими версиями. В связи с этим **возникает необходимость стандартизации различных свойств системы**. Наличие стандартов облегчает переносимость приложений между различными версиями *UNIX* и защищает как пользователей, так и производителей программного обеспечения. Поэтому в 80-х годах разработан ряд стандартов, оказывающих влияние на развитие *UNIX*. **Сейчас существуют десятки операционных систем, которые можно объединить под общим названием *UNIX***. В основном, это коммерческие версии, выпущенные производителями аппаратных платформ для компьютеров своего производства. Причины популярности *UNIX*:

- **Код системы написан на языке высокого уровня *C*, что сделало ее простой для понимания, изменения и переноса на другие платформы**. Можно смело сказать, что *UNIX* является одной из наиболее открытых систем.
- ***UNIX* — многозадачная многопользовательская система**. Один мощный сервер может обслуживать запросы большого количества пользователей. При этом необходимо администрирование только одной системы. Кроме того, система способна выполнять большое количество различных функций, в частности, работать, как вычислительный сервер, как сервер базы данных, как сетевой сервер, поддерживающий важнейшие сервисы сети и т.д.
- **Наличие стандартов**. Несмотря на разнообразие версий *UNIX*, **основой всего семейства являются принципиально одинаковая архитектура и ряд стандартных интерфейсов**. Для администратора переход на другую версию системы не составит большого труда, а для пользователей он может и вовсе оказаться незаметным.
- **Простой, но мощный модульный пользовательский интерфейс**. Имея в своем распоряжении набор утилит, каждая из которых решает узкую специализированную задачу, можно конструировать из них сложные комплексы.
- **Использование единой, легко обслуживаемой иерархической файловой системы**. Файловая система *UNIX* — это не только доступ к данным, хранящимся на диске. Через унифицированный интерфейс файловой системы осуществляется доступ к терминалам, принтерам, сети и т.п.
- **Очень большое количество приложений, в том числе свободно распространяемых**, начиная от простейших текстовых редакторов и заканчивая мощными системами управления базами данных.

(По материалам пособия "ОС UNIX", <http://base1.spiiras.nw.ru/univer/unix>)

Операционная система *Linux*

Начало созданию системы *Linux* положено в 1991 г. финским студентом *Линусом Торвальдсом* (*Linus Torvalds*). В сентябре 1991 года он распространил по e-mail первый прототип своей операционной системы, и призвал откликнуться на его работу всех, кому она нравится или нет. С этого момента многие программисты стали поддерживать *Linux*, добавляя драйверы устройств, разрабатывая разные продвинутое приложения и др. Атмосфера работы энтузиастов над полезным проектом, а также свободное распространение и использование исходных текстов стали основой феномена *Linux*. В настоящее время *Linux* — **очень мощная система**, но самое замечательное то, что **она бесплатная** (free).



Линус Торвальдс с символом *Linux* — "пингвином"

Линус Торвалдс разработал не саму операционную систему, а только ее ядро, подключив уже имеющиеся компоненты. Сторонние компании, увидев хорошие перспективы для развития своего бизнеса, довольно скоро стали насыщать ОС утилитами и прикладным ПО. **Недостаток такого подхода — отсутствие унифицированной и продуманной процедуры установки системы**, и это до сих пор является одним из главных сдерживающих факторов для более широкого распространения Linux.

Феномен *Linux* вызвал к жизни разговоры о том, что **родилась новая философия программирования**, принципиально отличающаяся от того, что было раньше. Традиционные стадии жизненного цикла программного продукта таковы: анализ требований, разработка спецификаций, проектирование, макетирование, написание исходного текста, отладка, документирование, тестирование и сопровождение. Главное, что отличает этот подход, — централизация управления разными стадиями и преимущественно "нисходящая" разработка (то есть постоянная детализация). Однако *Linux* создавалась по-иному. **Готовый работающий макет постоянно совершенствовался и развивался децентрализованной группой энтузиастов, действия которых лишь слегка координировались.** Налицо анархичный характер и "восходящая" разработка: сборка все более крупных блоков из ранее созданных мелких. Здесь можно отметить и другое. При традиционной разработке в основу кладется проектирование и написание текстов, при разработке по методу *Linux* — макетирование, отладка и тестирование. Первые два этапа распараллелить сложно, а с отладкой и тестированием дело обстоит полегче. Иными словами, **разработка по методу *Linux* — это метод проб и ошибок, построенный на интенсивном тестировании.** На любом этапе система должна работать, даже если это мини-версия того, к чему стремится разработчик. Естественный отбор оставляет только жизнеспособное. О том, что такое программирование — наука, искусство или ремесло, — спорят уже давно. И если в основе традиционной разработки ПО лежит прежде всего ремесло, то при разработке методом компьютерного дарвинизма — несомненно искусство.

Нетрудно заметить, что "восходящая" разработка характеризует так называемое исследовательское программирование, когда система строится вокруг ключевых компонентов и программ, которые создаются на ранних стадиях проекта, а затем постоянно модифицируются. **Отсутствие четкого плана, минимальное управление проектом, большое число сторонних территориально удаленных разработчиков, свободный обмен идеями и кодами — все это атрибуты нового программирования.** Об особенностях исследовательского программирования написано немало статей. Так, швейцарские профессора А.Киральф, К.Чен и Й.Нивергельт выделили следующие важные моменты:

- разработчик ясно представляет направление поиска, но не знает заранее, как далеко он сможет продвинуться к цели;
- нет возможности предвидеть объем ресурсов для достижения того или иного результата;
- разработка не поддается детальному планированию, она ведется методом проб и ошибок;
- такие работы связаны с конкретными исполнителями и отражают их личностные качества.

6.10. Что такое транслятор, компилятор, интерпретатор?

Транслятор (англ. *translator* — переводчик) — это программа-переводчик. Она преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Трансляторы реализуются в виде компиляторов или интерпретаторов. С точки зрения выполнения работы компилятор и интерпретатор существенно различаются.

Компилятор (англ. *compiler* — составитель, собиратель) читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

Интерпретатор (англ. *interpreter* — истолкователь, устный переводчик) переводит и выполняет программу *строка за строкой*.

После того, как программа откомпилирована, ни сама исходная программа, ни компилятор более не нужны. В то же время программа, обрабатываемая интерпретатором, должна заново *переводиться* на машинный язык при каждом очередном запуске программы.

Откомпилированные программы работают быстрее, но *интерпретируемые* проще исправлять и изменять.

Каждый конкретный язык ориентирован либо на компиляцию, либо на интерпретацию — в зависимости от того, для каких целей он создавался. Например, Паскаль обычно используется для решения довольно сложных задач, в которых важна скорость работы программ. Поэтому данный язык обычно реализуется с помощью *компилятора*.

С другой стороны, Бейсик создавался как язык для начинающих программистов, для которых построчное выполнение программы имеет неоспоримые преимущества.

Иногда для одного языка имеется *и компилятор, и интерпретатор*. В этом случае для разработки и тестирования программы можно воспользоваться интерпретатором, а затем откомпилировать отлаженную программу, чтобы повысить скорость ее выполнения.

6.11. Что такое системы программирования?

Система программирования — это система для разработки новых программ на конкретном языке программирования.

Современные системы программирования обычно предоставляют пользователям **мощные и удобные средства разработки программ**. В них входят:

- компилятор или интерпретатор;
- интегрированная среда разработки;

- средства создания и редактирования текстов программ;
- обширные [библиотеки стандартных программ](#) и функций;
- [отладочные программы](#), т.е. программы, помогающие находить и устранять ошибки в программе;
- "дружественная" к пользователю диалоговая среда;
- многооконный режим работы;
- мощные графические библиотеки; [утилиты](#) для работы с библиотеками
- встроенный [ассемблер](#);
- встроенная справочная служба;
- другие специфические особенности.

Популярные системы программирования — *Turbo Basic*, *Quick Basic*, *Turbo Pascal*, *Turbo C*.

В последнее время получили распространение системы программирования, ориентированные на создание *Windows-приложений*:



Borland Delphi 3.0

- пакет *Borland Delphi* (Дельфи) — блестящий наследник семейства компиляторов Borland Pascal, предоставляющий качественные и очень удобные средства визуальной разработки. Его исключительно быстрый компилятор позволяет эффективно и быстро решать практически любые задачи прикладного программирования.
- пакет *Microsoft Visual Basic* — удобный и популярный инструмент для создания Windows-программ с использованием визуальных средств. Содержит инструментарий для создания *диаграмм* и *презентаций*.
- пакет *Borland C++* — одно из самых распространённых средств для разработки DOS и Windows приложений.

Ниже для иллюстрации приведены на языках Бейсик, Паскаль и Си программы решения одной и той же простой задачи — вычисления суммы S элементов одномерного массива $A=(a_1, a_2, \dots, a_n)$.

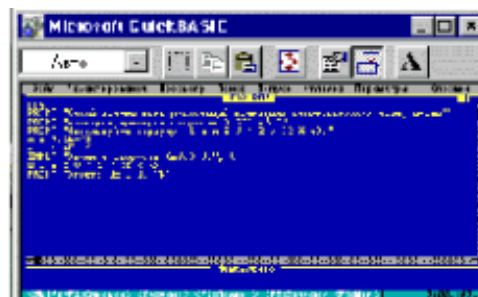
Язык Бейсик (BASIC — Beginners All-purpose Symbolic Instruction Code — универсальный символьный код для начинающих) был создан в 1965 г. **Дж. Кемени и Т.Курцем** как язык, облегчающий написание простых программ. Сейчас все чаще пишут **Basic** вместо **BASIC**, придавая другую трактовку названию: Basic — основной, базовый.

Существует много различных версий Бейсика — от очень простых до усовершенствованных, содержащих множество дополнительных языковых конструкций. Наибольшее распространение имеют следующие версии: **QuickBasic** (QBasic) 4.5 для DOS и **Visual Basic** 3.0-6.0 для Windows.

QuickBasic 4.5 фирмы Microsoft — это очень простой, но в то же время эффективный язык, унаследовавший от раннего Бейсика все его достоинства, но избавившийся от всех его недостатков и впитавший целый ряд передовых идей начала 90-х годов. Еще QuickBasic очень привлекателен своей средой программирования. Одна из удач — это использование своеобразного режима работы. Для быстрой работы в среде используется режим **интерпретатора**, а для окончательного перевода отлаженных программ на машинный язык используется **компилятор**. Из QBasic компилятор исключен. QBasic входит в минимальный комплект поставки программного обеспечения компьютера. Бейсик очень популярный язык программирования.

Программа на Бейсике

```
INPUT "N = "; N : DIM A(N)
FOR I = 1 TO N
    PRINT "A("; I; ") =";
    INPUT A(I)
NEXT I
S = 0
FOR I = 1 TO N
    S = S + A(I)
NEXT I
PRINT "Сумма ="; S
END
```



Окно среды программирования Quick Basic

Программа на Паскале

```
Program Summa;
Type Mas = Array [1 ..
100] of Real;
Var A : Mas;
    i, n: Integer;
    S : Real;
BEGIN
Write('n = '); ReadLn(n);
For i := 1 to n do
begin
Write('A[', i, '] = ');
ReadLn(A[i]);
end;

S := 0;
For i := 1 to n do
S := S + A[i];
WriteLn('S = ', S:8:2);
END.
```

Язык Паскаль был разработан в 1970 г. **Никласом Виртом** как язык обучения студентов программированию. Паскаль вырабатывает навыки соблюдения хорошего строгого стиля программирования, упрощающего разработку сложных программ.

Основные привлекательные черты Паскаля — **логичность, поддержка концепций структурного и процедурного программирования, работа с динамической памятью, возможность создания своих типов данных.** В Паскале программист должен всегда явно указывать, с какими конкретными переменными он желает работать и каковы типы этих переменных. Строгая типизация данных позволяет резко снизить количество ошибок, появляющихся в программе вследствие невнимательности или опечаток.

В своем первоначальном виде Паскаль имел довольно ограниченные возможности, но расширенный вариант этого языка — **Turbo Pascal**, является очень мощным языком программирования. Интегрированная оболочка Turbo Pascal, разработанная фирмой **Borland** (ныне Inprise), включающая в себя редактор, компилятор, компоновщик и отладчик, вместе с интерактивной справочной системой сделали разработку программ на Паскале делом простым и приятным.

Язык Си разработан **Деннисом Ритчи** в 1972 г. как язык, пригодный для программирования новой операционной системы UNIX.

Операционные системы ради повышения скорости работы традиционно писались на языке низкого уровня — ассемблере, но язык Си настолько хорошо зарекомендовал себя, что на нем было написано более 90% всего кода ОС UNIX. Язык СИ обрел популярность как так называемый язык среднего уровня, в котором удобство, краткость и мобильность языков высокого уровня сочетаются с возможностью непосредственного доступа к аппаратуре компьютера, что обычно достигаются только при программировании на языке Ассемблера.

Си не очень прост в изучении и требует тщательности в программировании, но позволяет создавать сложные и весьма эффективные программы.

Программа на СИ

```
# include <stdio.h>
# include <conio.h>
main()
{
float a[100], s; int i, n;
clrscr(); printf("n=");
scanf("%i", &n);
for (i = 1; i <= n; i++) {
printf("a[%i]=", i);
scanf("%f", &a[i]);}
s=0;
for (i = 1; i <= n; i++)
s = s + a[i];
printf("s = % f \ n", s);
return 0;
}
```


6.12. Для чего нужны инструментальные программы?

Инструментальные программные средства — это программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ.

По своему назначению они близки системам программирования. К инструментальным программам, например, относятся:

- редакторы;
- средства компоновки программ;
- отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
- вспомогательные программы, реализующие часто используемые системные действия;
- графические пакеты программ и т.п.

Инструментальные программные средства могут оказать помощь на всех стадиях разработки ПО.

6.13. Что такое текстовый редактор?

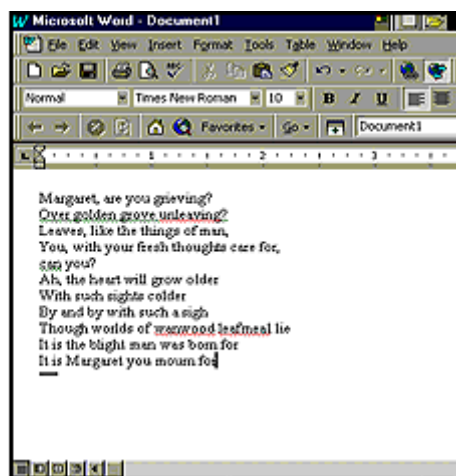
Текстовый редактор — это программа, используемая специально для ввода и редактирования текстовых данных.

Этими данными могут быть программа или какой-либо документ или же книга. Редактируемый текст выводится на экран, и пользователь может в диалоговом режиме вносить в него свои изменения.

Текстовые редакторы могут обеспечивать выполнение разнообразных функций, а именно:

- редактирование строк текста;
- возможность использования различных шрифтов символов;
- копирование и перенос части текста с одного места на другое или из одного документа в другой;
- контекстный поиск и замена частей текста;
- задание произвольных межстрочных промежутков;
- автоматический перенос слов на новую строку;
- автоматическая нумерация страниц;
- обработка и нумерация сносок;
- выравнивание краев абзаца;
- создание таблиц и построение диаграмм;
- проверка правописания слов и подбор синонимов;
- построение оглавлений и предметных указателей;
- распечатка подготовленного текста на принтере в нужном числе экземпляров и т.п.

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



Окно редактора Microsoft Word

Наиболее известный текстовый редактор — *Microsoft Word*.

Полнофункциональные издательские системы — *Microsoft Publisher*, *Corel Ventura* и *Adobe PageMaker*. Издательские системы незаменимы для компьютерной верстки и графики. Значительно облегчают работу с многостраничными документами, имеют возможности автоматической разбивки текста на страницы, расстановки номеров страниц, создания заголовков и т.д. Создание макетов любых изданий — от рекламных листовок до многостраничных книг и журналов — становится очень простым, даже для новичков.

6.14. Что такое графический редактор?

Графический редактор — это программа, предназначенная для автоматизации процессов построения на экране дисплея графических изображений. Предоставляет возможности рисования линий, кривых, раскраски областей экрана, создания надписей различными шрифтами и т.д.



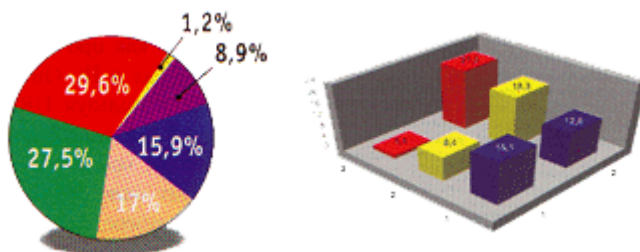
Рисунок, полученный с помощью Corel Draw!
[Увеличить изображение](#)

Большинство редакторов позволяют обрабатывать изображения, полученные с помощью *сканеров*, а также выводить картинки в таком виде, чтобы они могли быть включены в документ, подготовленный с помощью текстового редактора.

Некоторые редакторы позволяют получать изображения трёхмерных объектов, их сечений, разворотов, каркасных моделей и т.п.

Пользуется известностью *Corel DRAW!* — мощный графический редактор с функциями создания публикаций, снабжённый инструментами для редактирования графики и трёхмерного моделирования.

6.15. Каковы возможности систем деловой и научной графики?

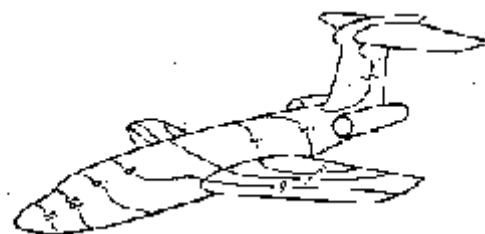


Круговые и столбиковые диаграммы

Системы деловой графики дают возможность выводить на экран различные виды графиков и диаграмм: >гистограммы; круговые и секторные диаграммы и т.д.

Эти системы позволяют наглядно представлять на экране различные данные и зависимости.

Системы научной и инженерной графики позволяют в цвете и в заданном масштабе отображать на экране следующие объекты:

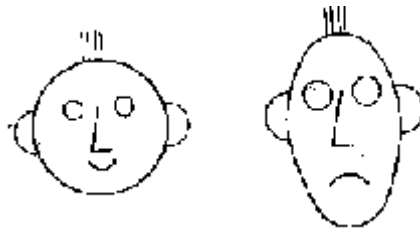


Чертеж с нанесенными изолиниями

- графики двумерных и трехмерных функций, заданных в табличном или аналитическом виде;
- системы изолиний, в том числе, и нанесённые на поверхность объекта;
- сечения, проекции, карты и т.д.

Для построения легко воспринимаемых реалистических изображений трёхмерных объектов системы инженерной графики позволяют удалять линии, не видимые наблюдателю.

Существуют **остроумные способы визуализации** наиболее простых **многомерных объектов** — множество точек. Один из них носит название "лица Чернова" (Чернов — современный американский математик).



"Лица Чернова"

Этим способом можно отображать 10-20-мерные множества. Суть способа такова: каждому из измерений сопоставляется один из параметров схематически изображённого человеческого лица, например, первое измерение дает отношение высоты лица к ширине, второе — размер носа, третье — расстояние между глазами и т.д.

Таким образом, каждой точке исходного множества будет сопоставлено лицо. Рассматривая эти лица, можно отобрать похожие между собой или же выделить абсолютно непохожие и тем самым произвести некую классификацию исходного множества.

6.16. Что такое табличный процессор?

Табличный процессор — это комплекс взаимосвязанных программ, предназначенный для обработки электронных таблиц.

Электронная таблица — это компьютерный эквивалент обычной таблицы, состоящей из строк и граф, на пересечении которых располагаются клетки, в которых содержится числовая информация, формулы или текст.

Значение в числовой клетке таблицы может быть либо записано, либо рассчитано по соответствующей формуле; в формуле могут присутствовать обращения к другим клеткам.

	A	B	C	D	E	F
1	Vertrod #1					
2	Foil Pouch 3630				Foil Pouch 3630	
3	Dwell #2	Dwell #3	Dwell #4	Dwell #5	Dwell #2	Dwell #5
4	1/30/95	1/30/95	1/30/95	1/30/95	1/31/95	1/31/95
5	17.85	22.9	23.65	24.8	19.15	23.1
6	19	26.4	22.6	27.5	18.6	21.1
7	13.15	25.45	26.9	25.9	19.01	24.1
8	19.05	24.55	27.65	28.2	19.25	24.1
9	14.1	25.4	29.1	21.45	17.65	22.1
10	27.6	21.45	27.4	23.55	22.02	21.1
11	36.5	21.2	23.3	20.65	21.9	26.1
12	24.25	26.3	24.8	20.5	19.75	23.1
13	15.05	23.55	23.6	23.9	14.65	22.1
14	24.9	25	24.45	25.1	17.9	22.1
15	26.55	27.35	20.25	24.1	15.85	21.1
16	30.7	22.45	24.55	22.95	15.7	21.1
17	31.3	26.35	25.6	22.3	13.25	24.1

Электронная таблица
Microsoft Excel

Каждый раз при изменении значения в клетке таблицы в результате записи в нее нового значения с клавиатуры *пересчитываются* также значения во всех тех клетках, в которых стоят величины, зависящие от данной клетки.

Графам и строкам можно присваивать наименования. Экран монитора трактуется как окно, через которое можно рассматривать таблицу целиком или по частям.

Табличные процессоры представляют собой удобное средство для проведения бухгалтерских и статистических расчетов. В каждом пакете имеются сотни встроенных математических функций и алгоритмов статистической обработки данных. Кроме того, имеются мощные средства для связи таблиц между собой, создания и редактирования электронных баз данных.

Специальные средства позволяют автоматически получать и распечатывать настраиваемые отчеты с использованием десятков различных типов таблиц, графиков, диаграмм, снабжать их комментариями и графическими иллюстрациями.

Табличные процессоры имеют встроенную справочную систему, предоставляющую пользователю информацию по конкретным командам меню и другие справочные данные. Многомерные таблицы позволяют быстро делать выборки в базе данных по любому критерию.

Самые популярные табличные процессоры — *Microsoft Excel (Эксель)* и *Lotus 1—2—3*.

В *Microsoft Excel* автоматизированы многие рутинные операции, специальные шаблоны помогают создавать отчёты, импортировать данные и многое другое.

Lotus 1—2—3 — профессиональный процессор электронных таблиц. Широкие графические возможности и удобный интерфейс пакета позволяют быстро ориентироваться в нём. С его помощью можно создать любой финансовый документ, отчёт для бухгалтерии, составить бюджет, а затем разместить все эти документы в базах данных.

6.17. Что такое системы управления базами данных?

База данных — это один или несколько файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации.

В базе данных предприятия, например, может храниться:

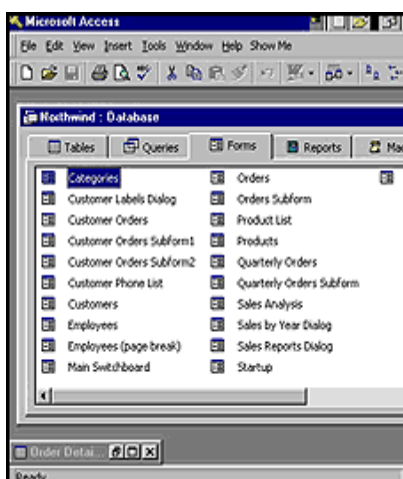
- вся информация о штатном расписании, о рабочих и служащих предприятия;
- сведения о материальных ценностях;
- данные о поступлении сырья и комплектующих;
- сведения о запасах на складах;
- данные о выпуске готовой продукции;
- приказы и распоряжения дирекции и т.п.

Даже небольшие изменения какой-либо информации могут приводить к значительным изменениям в разных других местах.

Пример. Издание приказа о повышении в должности одного работника приводит к изменениям не только в личном деле работника, но и к изменениям в списках подразделения, в котором он работает, в ведомостях на зарплату, в графике отпусков и т.п.

Базы данных используются под управлением *систем управления базами данных (СУБД)*.

Система управления базами данных (СУБД) — это система программного обеспечения, позволяющая обрабатывать обращения к базе данных, поступающие от прикладных программ конечных пользователей.



Окно СУБД Access

Системы управления базами данных позволяют *объединять* большие объемы информации и *обрабатывать* их, *сортировать*, *делать выборки* по определённым критериям и т.п.

Современные СУБД дают возможность включать в них не только *текстовую* и *графическую* информацию, но и звуковые фрагменты и даже видеоклипы.

Простота использования СУБД позволяет создавать новые базы данных, не прибегая к программированию, а пользуясь только встроенными функциями.

СУБД обеспечивают *правильность*, *полноту* и *непротиворечивость* данных, а также *удобный доступ* к ним.

Популярные СУБД — *FoxPro*, *Access for Windows*, *Paradox*.

Для менее сложных применений вместо СУБД используются **информационно-поисковые системы (ИПС)**, которые выполняют следующие функции:

- *хранение* большого объема информации;
- быстрый *поиск* требуемой информации;
- *добавление*, *удаление* и *изменение* хранимой информации;
- *вывод* ее в удобном для человека виде.

6.18. Что такое библиотеки стандартных подпрограмм?

Библиотека стандартных подпрограмм — это совокупность подпрограмм, составленных на одном из языков программирования и удовлетворяющих определенным единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм и т.п.

Стандартные подпрограммы имеют единую форму обращения, что обеспечивает простоту и удобство настройки параметров подпрограммы на решение конкретной задачи.

В качестве примера можно привести библиотеку стандартных подпрограмм по численным математическим методам решения уравнений, вычисления интегралов, нахождения экстремумов и т.п.

6.19. Что такое пакеты прикладных программ?

Пакеты прикладных программ (ППП) — это специальным образом организованные программные комплексы, рассчитанные на общее применение в определенной проблемной области и дополненные соответствующей технической документацией.

В зависимости от характера решаемых задач различают следующие разновидности ППП:

- пакеты для решения типовых инженерных, планово-экономических, общенаучных задач;
- пакеты системных программ;
- пакеты для обеспечения систем автоматизированного проектирования и систем автоматизации научных исследований;
- пакеты педагогических программных средств и другие.

Чтобы пользователь мог применить ППП для решения конкретной задачи, пакет должен обладать средствами настройки (иногда путём введения некоторых дополнений).

Каждый ППП обладает обычно рядом возможностей по методам обработки данных и формам их представления, полноте диагностики, что дает возможность пользователю выбрать подходящий для конкретных условий вариант.

ППП обеспечивают значительное снижение требований к уровню профессиональной подготовки пользователей в области программирования, вплоть до возможности эксплуатации пакета без программиста.

Часто пакеты прикладных программ располагают базами данных для хранения данных и передачи их прикладным программам.

6.20. Что такое интегрированные пакеты программ?

Интегрированные пакеты представляют собой набор нескольких программных продуктов, объединенных в единый удобный инструмент. Наиболее развитые из них включают в себя текстовый редактор, органайзер, электронную таблицу, СУБД, средства поддержки электронной почты, программу создания презентационной графики.

Результаты, полученные отдельными подпрограммами, могут быть объединены в окончательный документ, содержащий табличный, графический и текстовый материал.

Интегрированные пакеты, как правило, содержат некоторое ядро, обеспечивающее возможность тесного взаимодействия между составляющими.

Пример: **интегрированный пакет для написания книг, содержащих иллюстрации.** Он содержит:

- текстовый редактор;
- орфографический корректор на 80000 слов (программу обнаружения орфографических ошибок);
- программу слияния текстов;
- программу формирования оглавлений и составления указателей;
- автоматический поиск и замену слов и фраз;
- средства телекоммуникации;
- электронную таблицу;
- систему управления базами данных;
- модули графического оформления;
- графический редактор;
- возможность печати сотнями разных шрифтов и т.д.



Наиболее известные интегрированные пакеты:

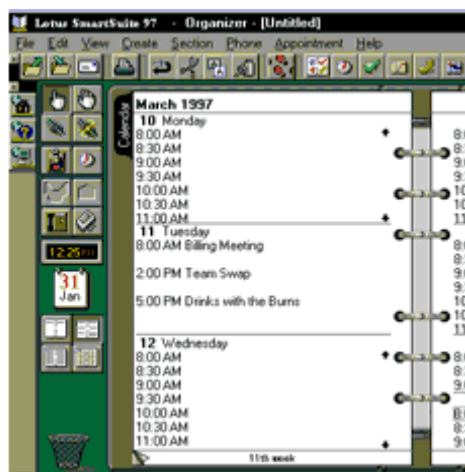
Microsoft Office. В этот мощный профессиональный пакет вошли такие необходимые программы, как текстовый редактор *WinWord*, электронная таблица *Excel*, программа создания презентаций *PowerPoint*, СУБД *Access*, средство поддержки электронной почты *Mail*. Мало того, все части этого пакета составляют единое целое, и даже внешне все программы выглядят единообразно, что облегчает как их освоение, так и ежедневное использование.

Microsoft Works — это очень простой и удобный пакет, объединяющий в себе текстовый редактор, электронные таблицы и базы данных, а также телекоммуникационные средства для соединения с другими компьютерами по

телефонным линиям. Пакет ориентирован на людей, не имеющих времени осваивать сложные продукты, на начинающих пользователей, а также на домашних пользователей.

6.21. Что такое органайзеры?

Органайзеры — это программы — электронные секретари.



Lotus Organizer

Они позволяют эффективно распорядиться рабочим временем, финансовыми средствами и т.п. Обладают возможностью автоматизации регулярных действий, составления персональных и групповых расписаний, планирования встреч, ведения записной книжки. В их состав традиционно входят календарь, часы, калькулятор и т.п.

Lotus Organizer — блокнот, разбитый по секциям: календарь, список дел, адресная и телефонная книга, планировщик, записная книжка, список памятных дат.

Microsoft Project позволяет спланировать проведение проектов и представить расписание в графическом виде, что очень удобно для сложных проектов.

6.22. Что такое сетевое программное обеспечение?

Сетевое программное обеспечение предназначено для организации совместной работы группы пользователей на разных компьютерах. Позволяет организовать общую файловую структуру, общие базы данных, доступные каждому члену группы. Обеспечивает возможность передачи сообщений и работы над общими проектами, возможность разделения ресурсов.

Функции и характеристики сетевых операционных систем (ОС)

(по материалам www-сайта Омского государственного технического университета
<http://edu.omgtu.omскеlecom.ru>)

К основным функциям сетевых ОС относят:

- управление каталогами и файлами;

- управление ресурсами;
- коммуникационные функции;
- защиту от несанкционированного доступа;
- обеспечение отказоустойчивости;
- управление сетью.

Управление каталогами и файлами в сетях заключается в обеспечении доступа к данным, физически расположенным в других узлах сети. Управление осуществляется с помощью специальной **сетевой файловой системы**. Файловая система позволяет обращаться к файлам путем применения привычных для локальной работы языковых средств. При обмене файлами должен быть обеспечен необходимый уровень конфиденциальности обмена (секретности данных).

Управление ресурсами включает обслуживание запросов на предоставление ресурсов, доступных по сети.

Коммуникационные функции обеспечивают адресацию, буферизацию, выбор направления для движения данных в разветвленной сети (маршрутизацию), управление потоками данных и др.

Защита от несанкционированного доступа — важная функция, способствующая поддержанию целостности данных и их конфиденциальности. Средства защиты могут разрешать доступ к определенным данным только с некоторых терминалов, в оговоренное время, определенное число раз и т.п. У каждого пользователя в корпоративной сети могут быть свои права доступа с ограничением совокупности доступных директорий или списка возможных действий, например, может быть запрещено изменение содержимого некоторых файлов.

Отказоустойчивость характеризуется сохранением работоспособности системы при воздействии дестабилизирующих факторов. Отказоустойчивость обеспечивается применением для серверов автономных источников питания, отображением или дублированием информации в дисковых накопителях. Под отображением обычно понимают наличие в системе двух копий данных с их расположением на разных дисках, но подключенных к одному контроллеру. Дублирование отличается тем, что для каждого из дисков с копиями используются разные контроллеры. Очевидно, что дублирование более надежно. Дальнейшее повышение отказоустойчивости связано с дублированием серверов, что однако требует дополнительных затрат на приобретение оборудования.

Управление сетью связано с применением соответствующих протоколов управления. Программное обеспечение управления сетью обычно состоит из менеджеров и агентов. **Менеджером** называется программа, вырабатывающая сетевые команды. **Агенты** представляют собой программы, расположенные в различных узлах сети. Они выполняют команды менеджеров, следят за состоянием узлов, собирают информацию о параметрах их функционирования, сигнализируют о происходящих событиях, фиксируют аномалии, следят за трафиком, осуществляют защиту от вирусов. Агенты с достаточной степенью интеллектуальности могут участвовать в восстановлении информации после сбоев, в корректировке параметров управления и т.п.

Программное обеспечение сетевых ОС распределено по узлам сети. Имеется ядро ОС, выполняющее большинство из охарактеризованных выше функций, и дополнительные программы (службы), ориентированные на реализацию протоколов верхних уровней, выполнение специфических функций для коммутационных серверов,

организацию распределенных вычислений и т.п. К сетевому программному обеспечению относят также **драйверы сетевых плат**. Для каждого типа ЛВС разработаны разные типы плат и драйверов. Внутри каждого типа ЛВС может быть много разновидностей плат с разными характеристиками интеллектуальности, скорости, объема буферной памяти.

В настоящее время наибольшее распространение получили три основные сетевые ОС — **UNIX, Windows NT и Novell Netware**.

ОС UNIX применяют преимущественно в крупных корпоративных сетях, поскольку эта система характеризуется высокой надежностью, возможностью легкого масштабирования сети. В Unix имеется ряд команд и поддерживающих их программ для работы в сети. Во-первых, это команды **ftp, telnet**, реализующие файловый обмен и эмуляцию удаленного узла на базе протоколов **TCP/IP**. Во-вторых, протокол, команды и программы **UUCP**, разработанные с ориентацией на асинхронную модемную связь по телефонным линиям между удаленными Unix-узлами в корпоративных и территориальных сетях.

ОС Windows NT включает серверную (**Windows NT Server**) и клиентскую (**Windows NT Workstation**) части и, тем самым, обеспечивает работу в сетях "клиент/сервер". Windows NT обычно применяют в средних по масштабам сетях.

ОС Novell Netware состоит из серверной части и оболочек **Shell**, размещаемых в клиентских узлах. Предоставляет пользователям возможность совместно использовать файлы, принтеры и другое оборудование. Содержит службу каталогов, общую распределённую базу данных пользователей и ресурсов сети. Эту ОС чаще применяют в небольших сетях.

6.23. Вопросы для самоконтроля.

- 6.1. Что такое программа?
- 6.2. Что включает в себя понятие "программное обеспечение"?
- 6.3. Назовите и характеризуйте основные категории программного обеспечения.
- 6.4. В чем отличие прикладных программ от системных и инструментальных?
- 6.5. Что входит в системное программное обеспечение?
- 6.6. В чем состоит назначение операционной системы?
- 6.7. Характеризуйте основные классы операционных систем.
- 6.8. Опишите процесс начальной загрузки операционной системы в оперативную память компьютера.
- 6.9. Что такое файл?
- 6.10. Как организована файловая система?
- 6.11. Какой модуль операционной системы осуществляет обслуживание файлов?

- 6.12.** Приведите пример иерархической файловой структуры.
- 6.13.** Что такое базовая система ввода-вывода (BIOS), и в каком разделе памяти она размещается?
- 6.14.** Из каких основных модулей состоит операционная система MS-DOS?
- 6.15.** Назовите основные разновидности программ-утилит и дайте им краткую характеристику.
- 6.16.** К каким категориям программного обеспечения относятся программные пакеты:
- Norton Commander;
 - MS-DOS;
 - Windows 3.x;
 - Windows-NT, Windows 95;
 - Microsoft Word;
 - Adobe PageMaker;
 - Turbo Pascal, Turbo Basic;
 - Microsoft Excel, Lotus;
 - FoxPro, Access for Windows;
 - Microsoft Office, Microsoft Works?
- 6.17.** Для чего предназначен пакет программ Norton Commander?
- 6.18.** Какой вид интерфейса удобнее для пользователя — командный или графический?
- 6.19.** Чем объясняется широкая популярность пакета Norton Commander?
- 6.20.** Что такое компьютерные вирусы, в чем состоят их вредные действия?
- 6.21.** Какие существуют средства борьбы с компьютерными вирусами?
- 6.22.** В чем суть процесса сжатия информации?
- 6.23.** Характеризуйте основные особенности операционных систем Windows-NT и Windows 95.
- 6.24.** Какие языки и системы программирования вы знаете и в чем их особенности?
- 6.25.** В чем отличие процесса интерпретации от процесса компиляции?
- 6.26.** Назовите основные функции текстовых редакторов.
- 6.27.** Какие дополнительные возможности редактирования текстов обеспечивают полнофункциональные издательские системы по сравнению с текстовыми редакторами?
- 6.28.** Назовите функциональные возможности табличного процессора.
- 6.29.** Какие виды входных данных могут быть введены в клетки электронных таблиц?

6.30. Дайте определение и опишите назначение базы данных.

6.31. Приведите пример возможного наполнения базы данных вашего учебного заведения.

6.32. Каковы основные функциональные возможности систем управления базами данных?

6.33. Что такое информационно-поисковые системы?

6.34. Дайте определение пакета прикладных программ.

6.35. Каково назначение библиотек стандартных программ?

6.36. Дайте определения интегрированного пакета программ.

6.37. Каково назначение сетевого программного обеспечения?



Глава 7. Алгоритмы. Алгоритмизация. Алгоритмические языки

7.1. Что такое алгоритм?

Понятие алгоритма такое же основополагающее для информатики, как и понятие информации. Именно поэтому важно в нем разобраться.

Название "алгоритм" произошло от латинской формы имени величайшего среднеазиатского математика **Мухаммеда ибн Муса ал-Хорезми** (Alhorithmi), жившего в 783—850 гг. В своей книге "Об индийском счете" он изложил правила записи натуральных чисел с помощью арабских цифр и правила действий над ними "столбиком", знакомые теперь каждому школьнику. В XII веке эта книга была переведена на латынь и получила широкое распространение в Европе.

Человек ежедневно встречается с необходимостью следовать тем или иным правилам, выполнять различные инструкции и указания. Например, переходя через дорогу на перекрестке без светофора надо сначала посмотреть направо. Если машин нет, то перейти полдороги, а если машины есть, ждать, пока они пройдут, затем перейти полдороги. После этого посмотреть налево и, если машин нет, то перейти дорогу до конца, а если машины есть, ждать, пока они пройдут, а затем перейти дорогу до конца.

В математике для решения типовых задач мы используем определенные правила, описывающие последовательности действий. Например, правила сложения дробных чисел, решения квадратных уравнений и т. д. Обычно любые инструкции и правила представляют собой последовательность действий, которые необходимо выполнить в определенном порядке. Для решения задачи надо знать, что дано, что следует получить и какие действия и в каком порядке следует для этого выполнить. Предписание, определяющее порядок выполнения действий над данными с целью получения искомого результатов, и есть алгоритм.

Алгоритм — заранее заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения решения задачи за конечное число шагов.

Это — не определение в математическом смысле слова, а, скорее, **описание** интуитивного понятия алгоритма, раскрывающее его сущность.

Понятие алгоритма является не только одним из главных понятий математики, но одним из главных понятий современной науки. Более того, с наступлением эры информатики алгоритмы становятся одним из важнейших факторов цивилизации [56].

7.2. Что такое "Исполнитель алгоритма"?

Исполнитель алгоритма — это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнять действия, предписываемые алгоритмом.

Исполнителя характеризуют:

- среда;
- элементарные действия;
- система команд;
- отказы.

Среда (или обстановка) — это "место обитания" исполнителя. Например, для исполнителя Робота из школьного учебника [1] среда — это бесконечное клеточное поле. Стены и закрашенные клетки тоже часть среды. А их расположение и положение самого Робота задают конкретное **состояние среды**.

Система команд. Каждый исполнитель может выполнять команды только из некоторого строго заданного списка — системы команд исполнителя. Для каждой команды должны быть заданы **условия применимости** (в каких состояниях среды может быть выполнена команда) и описаны **результаты выполнения команды**. Например, команда Робота "вверх" может быть выполнена, если выше Робота нет стены. Ее результат — смещение Робота на одну клетку вверх.

После вызова команды исполнитель совершает соответствующее **элементарное действие**.

Отказы исполнителя возникают, если команда вызывается при недопустимом для нее состоянии среды.

Обычно исполнитель ничего не знает о цели алгоритма. Он выполняет все полученные команды, не задавая вопросов "почему" и "зачем".

В информатике универсальным исполнителем алгоритмов является **компьютер**.

7.3. Какими свойствами обладают алгоритмы?

Основные свойства алгоритмов следующие:

1. Понятность для исполнителя — исполнитель алгоритма должен понимать, как его выполнять. Иными словами, имея алгоритм и произвольный вариант исходных данных, исполнитель должен знать, как надо действовать для выполнения этого алгоритма.

2. Дискретность (прерывность, раздельность) — алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).

3. Определенность — каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

4. Результативность (или конечность) состоит в том, что за конечное число шагов алгоритм либо должен приводить к решению задачи, либо после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.

5. Массовость означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

7.4. В какой форме записываются алгоритмы?

На практике наиболее распространены следующие формы представления алгоритмов:

- **словесная** (запись на естественном языке);
- **графическая** (изображения из графических символов);
- **псевдокоды** (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- **программная** (тексты на языках программирования).

7.5. Что такое словесный способ записи алгоритмов?

Словесный способ записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке.

Например. Записать алгоритм нахождения **наибольшего общего делителя (НОД)** двух натуральных чисел (алгоритм Эвклида).

Алгоритм может быть следующим:

1. задать два числа;
2. если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
3. определить большее из чисел;
4. заменить большее из чисел разностью большего и меньшего из чисел;
5. повторить алгоритм с шага 2.

Описанный алгоритм применим к любым натуральным числам и должен приводить к решению поставленной задачи. Убедитесь в этом самостоятельно, определив с помощью этого алгоритма наибольший общий делитель чисел 125 и 75.

Словесный способ не имеет широкого распространения, так как такие описания:

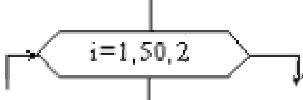
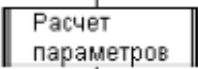
- строго не формализуемы;
- страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний.

7.6. Что такое графический способ записи алгоритмов?

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным.

При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.

Такое графическое представление называется схемой алгоритма или **блок-схемой**. В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде **блочного символа**. Блочные символы соединяются **линиями переходов**, определяющими очередность выполнения действий. В таблице приведены наиболее часто употребляемые символы.

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме, стандартной подпрограмме
Ввод-вывод		Ввод-вывод в общем виде
Пуск-останов		Начало, конец алгоритма, вход и выход в подпрограмму
Документ		Вывод результатов на печать

Блок "**процесс**" применяется для обозначения действия или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединять в один блок. Представление отдельных операций достаточно свободно.

Блок "**решение**" используется для обозначения переходов управления по условию. В каждом блоке "решение" должны быть указаны вопрос, условие или сравнение, которые он определяет.

Блок "**модификация**" используется для организации циклических конструкций. (Слово модификация означает видоизменение, преобразование). Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и шаг изменения значения параметра для каждого повторения.

Блок "**предопределенный процесс**" используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращений к библиотечным подпрограммам.

7.7. Что такое псевдокод?

Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов.

Псевдокод занимает промежуточное место между естественным и формальным языками. С одной стороны, он близок к обычному естественному языку, поэтому алгоритмы могут на нем записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются некоторые формальные конструкции и математическая символика, что приближает запись алгоритма к общепринятой математической записи.

В псевдокоде не приняты строгие синтаксические правила для записи команд, присущие формальным языкам, что облегчает запись алгоритма на стадии его проектирования и дает возможность использовать более широкий набор команд, рассчитанный на абстрактного исполнителя.

Однако в псевдокоде обычно **имеются некоторые конструкции, присущие формальным языкам**, что облегчает переход от записи на псевдокоде к записи алгоритма на формальном языке. В частности, в псевдокоде, так же, как и в формальных языках, есть служебные слова, смысл которых определен раз и навсегда. Они выделяются в печатном тексте жирным шрифтом, а в рукописном тексте подчеркиваются.

Единого или формального определения псевдокода не существует, поэтому возможны различные псевдокоды, отличающиеся набором служебных слов и основных (базовых) конструкций.

Примером псевдокода является школьный алгоритмический язык в русской нотации (школьный АЯ), описанный в учебнике А.Г. Кушниренко и др. "Основы информатики и вычислительной техники", 1991. Этот язык в дальнейшем мы будем называть просто "алгоритмический язык".

7.8. Как записываются алгоритмы на школьном алгоритмическом языке?

Основные служебные слова

алг (алгоритм)	сим (символьный)	дано	для	да
арг (аргумент)	лит (литерный)	надо	от	нет
рез (результат)	лог (логический)	если	до	при
нач (начало)	таб (таблица)	то	знач	выбор
кон (конец)	нц (начало цикла)	иначе	и	ввод
цел (целый)	кц (конец цикла)	все	или	вывод
вещ (вещественный)	длин (длина)	пока	не	утв

Общий вид алгоритма:

```
алг название алгоритма (аргументы и результаты)
  дано условия применимости алгоритма
  надо цель выполнения алгоритма
  нач описание промежуточных величин
  | последовательность команд (тело алгоритма)
кон
```

Часть алгоритма от слова **алг** до слова **нач** называется **заголовком**, а часть, заключенная между словами **нач** и **кон** — **телом** алгоритма.

В предложении **алг** после названия алгоритма в круглых скобках указываются характеристики (**арг**, **рез**) и тип значения (**цел**, **вещ**, **сим**, **лит** или **лог**) всех **входных** (аргументы) и **выходных** (результаты) переменных. При описании массивов (таблиц) используется служебное слово **таб**, дополненное **граничными парами** по каждому индексу элементов массива.

Примеры предложений **алг**:

```
алг Объем и площадь цилиндра ( арг вещ R, H, рез вещ V, S )
алг Корни КвУр ( арг вещ a, b, c, рез вещ x1, x2, рез лит t )
алг Исключить элемент ( арг цел N, арг рез вещ таб A[1:N] )
алг Диагональ ( арг цел N, арг цел таб A[1:N, 1:N], рез лит Ответ )
```

Предложения **дано** и **надо** не обязательны. В них рекомендуется записывать утверждения, описывающие состояние среды исполнителя алгоритма, например:

1. **алг** Замена (**арг** **лит** Str1, Str2, **арг** **рез** **лит** Text)
2. **дано** | длины подстрок Str1 и Str2 совпадают
3. **надо** | всюду в строке Text подстрока Str1 заменена на Str2
- 4.
5. **алг** Число максимумов (**арг** **цел** N, **арг** **вещ** **таб** A[1:N], **рез** **цел** K)
6. **дано** | N>0
7. **надо** | K — число максимальных элементов в таблице A
- 8.
9. **алг** Сопротивление (**арг** **вещ** R1, R2, **арг** **цел** N, **рез** **вещ** R)
10. **дано** | N>5, R1>0, R2>0
11. **надо** | R — сопротивление схемы

Здесь в предложениях **дано** и **надо** после знака "|" записаны **комментарии**. Комментарии можно помещать в конце любой строки. Они не обрабатываются транслятором, но существенно облегчают понимание алгоритма.

Команды школьного АЯ

Команда присваивания. Служит для вычисления выражений и присваивания их значений переменным. Общий вид: **A := B**, где знак " := " означает команду **заменить** **прежнее значение переменной, стоящей в левой части, на вычисленное значение выражения, стоящего в правой части.**

Например, `a := (b+c) * sin(Pi/4); i := i+1.`

Команды ввода и вывода.

- **ввод** имени переменных
- **вывод** имени переменных, выражения, тексты.

Команды если и выбор. Применяют для организации [ветвлений](#).

Команды для и пока. Применяют для организации [циклов](#).

Пример записи алгоритма на школьном АЯ

```
алг Сумма квадратов (арг цел n, рез цел S)
  дано | n > 0
  надо | S = 1*1 + 2*2 + 3*3 + ... + n*n
нач цел i
  ввод n; S:=0
  нц для i от 1 до n
    S:=S+i*i
  кц
  вывод "S = ", S
кон
```

7.9. Что такое базовые алгоритмические структуры?

Алгоритмы можно представлять как некоторые структуры, состоящие из отдельных **базовых** (т.е. основных) **элементов**. Естественно, что при таком подходе к алгоритмам изучение основных принципов их конструирования должно начинаться с изучения этих базовых элементов. Для их описания будем использовать язык схем алгоритмов и школьный алгоритмический язык.

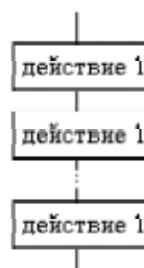
Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур: следование, ветвление, цикл.

Характерной особенностью базовых структур является наличие в них **одного входа и одного выхода**.

1. Базовая структура "следование". Образуется последовательностью действий, следующих одно за другим:

действие 1
 действие 2

 действие n



2. Базовая структура "ветвление". Обеспечивает в зависимости от результата проверки условия (**да** или **нет**) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к **общему выходу**, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран. Структура **ветвление** существует в четырех основных вариантах:

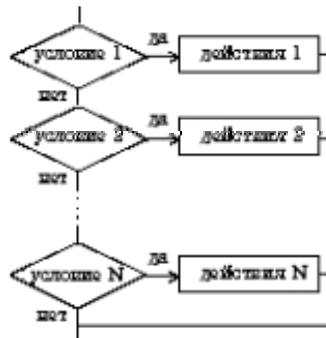
- если—то;
- если—то—иначе;
- выбор;
- выбор—иначе.

Школьный алгоритмический язык	Язык блок-схем
1. если—то	
<pre> если условие то действия все </pre>	
2. если—то—иначе	
<pre> если условие то действия 1 иначе действия 2 все </pre>	

3. выбор

выбор
при условие 1: действия 1
при условие 2: действия 2

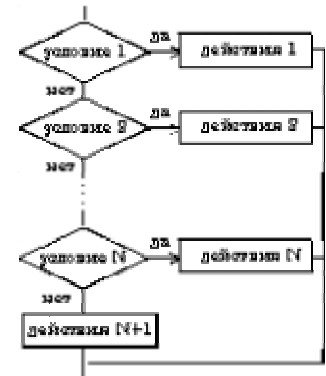
при условие N: действия N
все



4. выбор—иначе

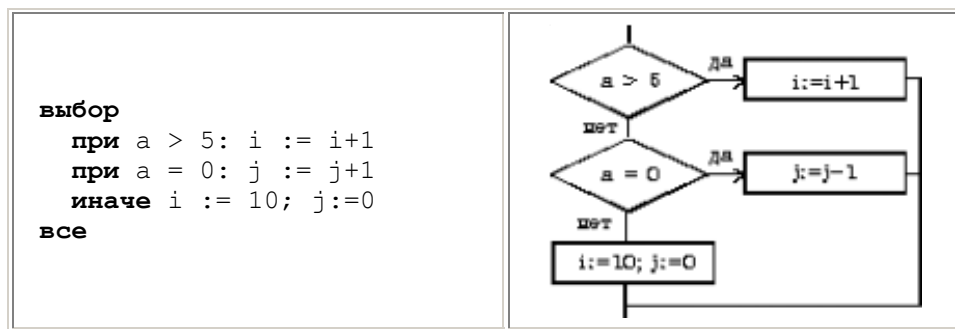
выбор
при условие 1: действия 1
при условие 2: действия 2

при условие N: действия N
иначе действия N+1
все



Примеры структуры **ветвление**

Школьный алгоритмический язык	Язык блок-схем
<pre> если x > 0 то y := sin(x) все </pre>	
<pre> если a > b то a := 2*a; b := 1 иначе b := 2*b все </pre>	
<pre> выбор при n = 1: y := sin(x) при n = 2: y := cos(x) при n = 3: y := 0 все </pre>	



3. Базовая структура "цикл". Обеспечивает многократное выполнение некоторой совокупности действий, которая называется телом цикла. Основные разновидности циклов представлены в таблице:

Школьный алгоритмический язык	Язык блок-схем
<p>Цикл типа пока. Предписывает выполнять тело цикла до тех пор, пока выполняется условие, записанное после слова пока.</p>	
<pre> нц пока условие тело цикла (последовательность действий) кц </pre>	
<p>Цикл типа для. Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне.</p>	
<pre> нц для i от i1 до i2 тело цикла (последовательность действий) кц </pre>	

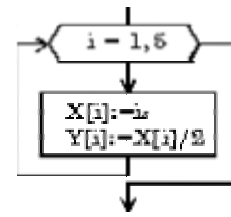
Примеры структуры **ЦИКЛ**

Школьный алгоритмический язык	Язык блок-схем
<pre> нц пока i <= 5 S := S+A[i] i := i+1 кц </pre>	

```

нц для i от 1 до 5
  X[i] := i*i*i
  Y[i] := X[i]/2
кц

```



7.10. Какие циклы называют итерационными?

Особенностью итерационного цикла является то, что число повторений операторов тела цикла заранее неизвестно. Для его организации используется цикл типа **пока**. Выход из итерационного цикла осуществляется в случае выполнения заданного условия.

На каждом шаге вычислений происходит **последовательное приближение к искомому результату и проверка условия достижения последнего**.

Пример. Составить алгоритм вычисления бесконечной суммы

$$S = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots (-1)^{i-1} \frac{x^i}{i} + \dots$$

с заданной точностью ε (для данной знакопеременной бесконечной суммы требуемая точность будет достигнута, когда очередное слагаемое станет по абсолютной величине меньше ε).

Вычисление сумм — типичная циклическая задача. Особенностью же нашей конкретной задачи является то, что число слагаемых (а, следовательно, и число повторений тела цикла) заранее неизвестно. Поэтому выполнение цикла должно завершиться в момент достижения требуемой точности.

При составлении алгоритма нужно учесть, что знаки слагаемых чередуются и степень числа x в числителях слагаемых возрастает.

Решая эту задачу "в лоб" путем вычисления на каждом i -ом шаге частичной суммы

$$S := S + ((-1)**(i-1)) * (x**i) / i ,$$

мы получим очень неэффективный алгоритм, требующий выполнения большого числа операций. Гораздо лучше организовать вычисления следующим образом: если обозначить числитель какого-либо слагаемого буквой p , то у следующего слагаемого числитель будет равен $-p*x$ (знак минус обеспечивает чередование знаков слагаемых), а само слагаемое m будет равно p/i , где i — номер слагаемого.

Сравните эти два подхода по числу операций.

Алгоритм на школьном АЯ	Блок-схема алгоритма
<pre> алг Сумма (арг вещ x, Eps, рез вещ S) дано 0 < x < 1 надо S = x - x**2/2 + x**3/3 - ... нач цел i, вещ m, p ввод x, Eps S := 0; i := 1 начальные значения m := 1; p := -1 нц пока abs(m) > Eps p := -p*x p - числитель очередного слагаемого m := p/i m - очередное слагаемое S := S + m S - частичная сумма i := i + 1 i - номер очередного слагаемого кц вывод S кон </pre>	<pre> graph TD Start([начало]) --> Input[/x, ε/] Input --> Init[S := 0; i := 1 m := 1; p := -1] Init --> Decision{ m > ε } Decision -- да --> Loop[S := S+m; i := i+1 p := -p*x; m := p/i] Loop --> Decision Decision -- нет --> Output[/S/] Output --> End([конец]) </pre>

Алгоритм, в состав которого входит итерационный цикл, называется **итерационным алгоритмом**. Итерационные алгоритмы используются при реализации итерационных численных методов.

В итерационных алгоритмах необходимо обеспечить обязательное достижение условия выхода из цикла (**сходимость итерационного процесса**). В противном случае произойдет "**зацикливание**" алгоритма, т.е. не будет выполняться основное свойство алгоритма — **результативность**.

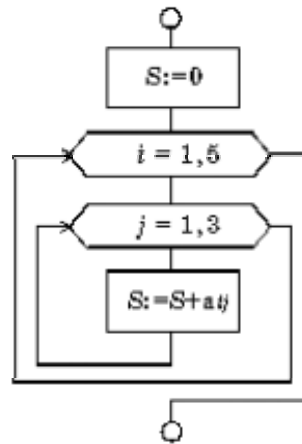
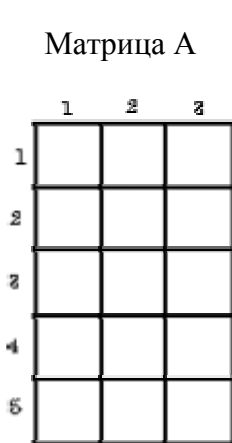
7.11. Что такое вложенные циклы?

Возможны случаи, когда внутри тела цикла необходимо повторять некоторую последовательность операторов, т. е. организовать внутренний цикл. Такая структура получила название **цикла в цикле** или **вложенных циклов**. Глубина вложения циклов (то есть количество вложенных друг в друга циклов) может быть различной.

При использовании такой структуры для экономии машинного времени необходимо выносить из внутреннего цикла во внешний все операторы, которые не зависят от параметра внутреннего цикла.

Пример вложенных циклов **ДЛЯ**

Вычислить сумму элементов заданной матрицы A(5,3).

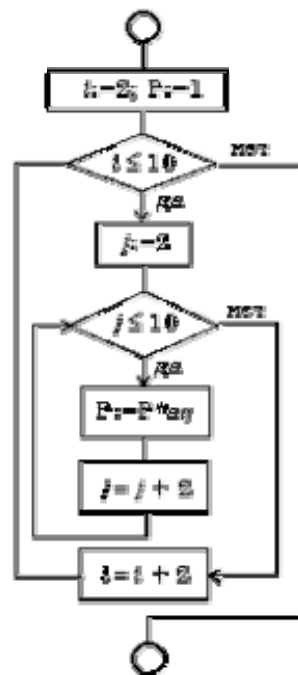


```

S := 0;
нц для i от 1 до 5
  нц для j от 1 до 3
    S:=S+A[i,j]
  кц
кц
  
```

Пример вложенных циклов пока

Вычислить произведение тех элементов заданной матрицы A(10,10), которые расположены на пересечении четных строк и четных столбцов.



```

i:=2; P:=1
нц пока i <= 10
  j:=2
  нц пока j <= 10
    P:=P*A[i,j]
    j:=j+2
  кц
  i:=i+2
кц
  
```

7.12. Чем отличается программный способ записи алгоритмов от других?

При записи алгоритма в словесной форме, в виде блок-схемы или на псевдокоде допускается определенный произвол при изображении команд. Вместе с тем такая запись точна настолько, что позволяет человеку понять суть дела и исполнить алгоритм.

Однако на практике в качестве исполнителей алгоритмов используются специальные автоматы — компьютеры. Поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на понятном ему языке. И здесь на первый план

выдвигается необходимость точной записи команд, не оставляющей места для произвольного толкования их исполнителем.

Следовательно, **язык для записи алгоритмов должен быть формализован**. Такой язык принято называть **языком программирования**, а запись алгоритма на этом языке — **программой для компьютера**.

7.13. Что такое уровень языка программирования?

В настоящее время в мире существует несколько сотен реально используемых языков программирования. Для каждого есть своя область применения.

Любой алгоритм, как мы знаем, 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 уровни языков программирования:

- [машинные](#);
- машинно-ориентированные ([ассемблеры](#));
- машинно-независимые (языки высокого уровня).

Машинные языки и машинно-ориентированные языки — это языки **низкого уровня**, требующие указания мелких деталей процесса обработки данных. Языки же **высокого уровня** имитируют естественные языки, используя некоторые слова разговорного языка и общепринятые математические символы. Эти языки более удобны для человека.

Языки высокого уровня делятся на:

- **процедурные (алгоритмические)** (Basic, Pascal, C и др.), которые предназначены для однозначного описания алгоритмов; для решения задачи процедурные языки требуют в той или иной форме явно записать процедуру ее решения;
- **логические** (Prolog, Lisp и др.), которые ориентированы не на разработку алгоритма решения задачи, а на систематическое и формализованное описание задачи с тем, чтобы решение следовало из составленного описания;
- **объектно-ориентированные** (Object Pascal, C++, Java и др.), в основе которых лежит **понятие объекта, сочетающего в себе данные и действия над ними**. Программа на объектно-ориентированном языке, решая некоторую задачу, по сути описывает часть мира, относящуюся к этой задаче. Описание действительности в форме системы взаимодействующих объектов естественнее, чем в форме взаимодействующих процедур.

7.14. Какие у машинных языков достоинства и недостатки?

Каждый компьютер имеет свой машинный язык, то есть свою совокупность машинных команд, которая отличается количеством адресов в команде, назначением информации, задаваемой в адресах, набором операций, которые может выполнить машина и др.

При программировании на машинном языке программист может держать под своим контролем каждую команду и каждую ячейку памяти, использовать все возможности имеющихся машинных операций.

Но процесс написания программы на машинном языке очень **трудоемкий и утомительный**. Программа получается **громоздкой, труднообозримой, ее трудно отлаживать, изменять и развивать**.

Поэтому в случае, когда нужно иметь эффективную программу, в максимальной степени учитывающую специфику конкретного компьютера, вместо машинных языков используют близкие к ним машинно-ориентированные языки (ассемблеры).

7.15. Что такое язык ассемблера?

Язык ассемблера — это машинно-зависимый язык низкого уровня, в котором короткие мнемонические имена соответствуют отдельным машинным командам. Используется для представления в удобочитаемой форме программ, записанных в машинном коде.

Язык ассемблера позволяет программисту пользоваться **текстовыми мнемоническими** (то есть легко запоминаемыми человеком) **кодами**, по своему усмотрению **присваивать символические имена регистрам компьютера и памяти, а также задавать удобные для себя способы адресации**. Кроме того, он позволяет использовать различные системы счисления (например, десятичную или шестнадцатеричную) для представления числовых констант, использовать в программе комментарии и др.

Программы, написанные на языке ассемблера, требуют значительно меньшего объема памяти и времени выполнения. Знание программистом языка ассемблера и машинного кода дает ему понимание архитектуры машины. Несмотря на то, что большинство специалистов в области программного обеспечения разрабатывают программы на языках высокого уровня, таких, как Object Pascal или C, наиболее мощное и эффективное программное обеспечение полностью или частично написано на языке ассемблера.

Языки высокого уровня были разработаны для того, чтобы освободить программиста от учета технических особенностей конкретных компьютеров, их архитектуры. В противоположность этому, язык ассемблера разработан с целью учесть конкретную специфику процессора. Следовательно, для того, чтобы написать программу на языке ассемблера для конкретного компьютера, важно знать его архитектуру [57].

В качестве примера приведем программу на языке ассемблера для IBM PC. Программа вычисляет значение $a = b + c$ для целых a , b и c :

	.MODEL SMALL	Определяет модель памяти. В данном случае модель SMALL.
	.DATA	Начало сегмента данных.
b	DW 5	Определяет длину слова (DW - double word) и количество слов (5).
c	DW 3	Определяет длину слова (DW - double word) и количество слов (3).
a	DW ?	Определяет длину слова (DW - double word) и количество слов (?).
	.CODE	Начало сегмента кода.
begin	MOV AX,@DATA	Загружает адрес начала сегмента данных (DATA) в регистр AX.
	MOV DS,AX	Загружает адрес сегмента данных (DATA) в регистр DS (Data Segment).
	MOV AX,B	Загружает значение переменной B в регистр AX.
	ADD AX,C	Добавляет значение переменной C к значению в регистре AX.
	MOV A,AX	Загружает значение из регистра AX в переменную A.
	MOV AH,4CH	Загружает значение 4CH в регистр AH.
	INT 21H	Вызывает прерывание 21H.
	END begin	Конец программы.

Перевод программы с языка ассемблера на машинный язык осуществляется специальной программой, которая называется **ассемблером** и является, по сути, простейшим **транслятором**.

7.16. В чем преимущества алгоритмических языков перед машинными?

Основные преимущества таковы:

- **алфавит** алгоритмического языка значительно шире алфавита машинного языка, что существенно повышает наглядность текста программы;
- **набор операций**, допустимых для использования, не зависит от набора машинных операций, а выбирается из соображений удобства формулирования алгоритмов решения задач определенного класса;
- **формат предложений** достаточно гибок и удобен для использования, что позволяет с помощью одного предложения задать достаточно содержательный этап обработки данных;
- требуемые операции задаются с помощью **общепринятых математических обозначений**;
- **данным в алгоритмических языках присваиваются индивидуальные имена**, выбираемые программистом;
- в языке может быть предусмотрен значительно **более широкий набор типов данных** по сравнению с набором машинных типов данных.

Таким образом, алгоритмические языки в значительной мере являются **машинно-независимыми**. Они облегчают **работу программиста** и **повышают надежность создаваемых программ**.

7.17. Какие компоненты образуют алгоритмический язык?

Алгоритмический язык (как и любой другой язык) образуют три его составляющие: *алфавит, синтаксис и семантика.*

Алфавит — это фиксированный для данного языка набор основных символов, т.е. "букв алфавита", из которых должен состоять любой текст на этом языке — никакие другие символы в тексте не допускаются.

Синтаксис — это правила построения фраз, позволяющие определить, правильно или неправильно написана та или иная фраза. Точнее говоря, синтаксис языка представляет собой набор правил, устанавливающих, какие комбинации символов являются осмысленными предложениями на этом языке.

Семантика определяет смысловое значение предложений языка. Являясь системой правил истолкования отдельных языковых конструкций, семантика устанавливает, какие последовательности действий описываются теми или иными фразами языка и, в конечном итоге, какой алгоритм определен данным текстом на алгоритмическом языке.

7.18. Какие понятия используют алгоритмические языки?

Каждое понятие алгоритмического языка подразумевает некоторую синтаксическую единицу (конструкцию) и определяемые ею свойства программных объектов или процесса обработки данных.

Понятие языка определяется во взаимодействии синтаксических и семантических правил. Синтаксические правила показывают, как образуется данное понятие из других понятий и букв алфавита, а семантические правила определяют свойства данного понятия

Основными понятиями в алгоритмических языках обычно являются следующие.

1. Имена (идентификаторы) — употребляются для обозначения объектов программы (переменных, массивов, функций и др.).

2. Операции. Типы операций:

- арифметические операции $+$, $-$, $*$, $/$ и др.;
- логические операции **и**, **или**, **не**;
- операции отношения $<$, $>$, \leq , \geq , $=$, \diamond ;
- операция сцепки (иначе, "присоединения", "конкатенации") символьных значений друг с другом с образованием одной длинной строки; изображается знаком "+".

3. Данные — величины, обрабатываемые программой. Имеется три основных вида данных: константы, переменные и массивы.

- **Константы** — это данные, которые зафиксированы в тексте программы и не изменяются в процессе ее выполнения.

Примеры констант:

- **числовые** 7.5 , 12 ;
 - **логические** да (истина), нет (ложь);
 - **символьные** (содержат ровно один символ) "А" , "+" ;
 - **литерные** (содержат произвольное количество символов) "а0", "Мир", "" (пустая строка).
- **Переменные** обозначаются именами и могут изменять свои значения в ходе выполнения программы. Переменные бывают **целые, вещественные, логические, символьные и литерные**.
 - **Массивы** — последовательности **однотипных элементов, число которых фиксировано и которым присвоено одно имя**. Положение элемента в массиве однозначно определяется его индексами (одним, в случае одномерного массива, или несколькими, если массив многомерный). Иногда массивы называют **таблицами**.

4. Выражения — предназначены для выполнения **необходимых вычислений**, состоят из констант, переменных, указателей функций (например, $\exp(x)$), объединенных знаками операций.

Выражения записываются в виде **линейных последовательностей символов** (без подстрочных и надстрочных символов, "многоэтажных" дробей и т.д.), что позволяет вводить их в компьютер, последовательно нажимая на соответствующие клавиши клавиатуры.

Различают выражения **арифметические, логические и строковые**.

- **Арифметические выражения** служат для определения одного числового значения. Например, $(1+\sin(x))/2$. Значение этого выражения при $x=0$ равно 0.5, а при $x=\pi/2$ — единице.
- **Логические выражения** описывают некоторые условия, которые могут удовлетворяться или не удовлетворяться. Таким образом, логическое выражение может принимать только два значения — **"истина"** или **"ложь"** (да или нет). Рассмотрим в качестве примера логическое выражение $x*x + y*y < r*r$, определяющее принадлежность точки с координатами (x, y) внутренней области круга радиусом r с центром в начале координат. При $x=1, y=1, r=2$ значение этого выражения — **"истина"**, а при $x=2, y=2, r=1$ — **"ложь"**.
- **Строковые (литерные) выражения, значениями которых являются тексты**. В строковые выражения могут входить литерные и строковые константы, литерные и строковые переменные, литерные функции, разделенные знаками операции сцепки. Например, $A + B$ означает присоединение строки B к концу строки A . Если $A = \text{"куст"}$, а $B = \text{"зеленый"}$, то значение выражения $A + B$ есть **"куст зеленый"**.

5. Операторы (команды). Оператор — это наиболее крупное и содержательное понятие языка: **каждый оператор представляет собой законченную фразу языка и**

определяет некоторый вполне законченный этап обработки данных. В состав операторов входят:

- ключевые слова;
- данные;
- выражения и т.д.

Операторы подразделяются на исполняемые и неисполняемые. **Неисполняемые** операторы предназначены для описания данных и структуры программы, а **исполняемые** — для выполнения различных действий (например, оператор присваивания, операторы ввода и вывода, условный оператор, операторы цикла, оператор процедуры и др.).

7.19. Что такое стандартная функция?

При решении различных задач с помощью компьютера бывает необходимо вычислить логарифм или модуль числа, синус угла и т.д.

Вычисления часто употребляемых функций осуществляются посредством подпрограмм, называемых стандартными функциями, которые заранее запрограммированы и встроены в транслятор языка.

Таблица стандартных функций школьного алгоритмического языка

Название и математическое обозначение функции		Указатель функции
Абсолютная величина (модуль)	$ x $	abs(x)
Корень квадратный	\sqrt{x}	sqrt(x)
Натуральный логарифм	$\ln x$	ln(x)
Десятичный логарифм	$\lg x$	lg(x)
Экспонента (степень числа $e \sim 2.72$)	e^x	exp(x)
Знак числа x (- 1, если $x < 0$; 0, если $x = 0$; 1, если $x > 0$)	sign x	sign(x)
Целая часть x (т.е. максимальное целое число, не превосходящее x)		int(x)
Минимум из чисел x и y		min(x,y)
Максимум из чисел x и y		max(x,y)
Частное от деления целого x на целое y		div(x,y)
Остаток от деления целого x на целое y		mod(x,y)
Случайное число в диапазоне от 0 до $x - 1$		rnd(x)
Синус (угол в радианах)	$\sin x$	sin(x)
Косинус (угол в радианах)	$\cos x$	cos(x)

Название и математическое обозначение функции		Указатель функции
Тангенс (угол в радианах)	tg x	tg(x)
Котангенс (угол в радианах)	ctg x	ctg(x)
Арксинус (главное значение в радианах)	arcsin x	arcsin(x)
Арккосинус (главное значение в радианах)	arccos x	arccos(x)
Арктангенс (главное значение в радианах)	arctg x	arctg(x)
Арккотангенс (главное значение в радианах)	arcctg x	arcctg(x)

В качестве аргументов функций можно использовать константы, переменные и выражения. Например:

$\sin(3.05)$ $\sin(x)$ $\sin(2 * y + t / 2)$ $\sin((\exp(x) + 1) ** 2)$
 $\min(a, 5)$ $\min(a, b)$ $\min(a + b, a * b)$ $\min(\min(a, b), \min(c, d))$

Каждый язык программирования имеет свой набор стандартных функций.

7.20. Как записываются арифметические выражения?

Арифметические выражения записываются по следующим правилам:

- Нельзя опускать знак умножения между сомножителями и ставить рядом два знака операций.
- Индексы элементов массивов записываются в квадратных (школьный АЯ, Pascal) или круглых (Basic) скобках.
- Для обозначения переменных используются буквы латинского алфавита.
- **Операции выполняются в порядке старшинства:** сначала вычисление функций, затем возведение в степень, потом умножение и деление и в последнюю очередь — сложение и вычитание.
- **Операции одного старшинства выполняются слева направо.** Однако, в школьном АЯ есть одно **исключение** из этого правила: операции возведения в степень выполняются справа налево. Так, выражение $2**(3**2)$ в школьном АЯ вычисляется как $2**(3**2) = 512$. В языке QBasic аналогичное выражение 2^3^2 вычисляется как $(2^3)^2 = 64$. А в языке **Pascal** вообще не предусмотрена операция возведения в степень, в Pascal x^y записывается как $\exp(y*\ln(x))$, а x^y^z как $\exp(\exp(z*\ln(y))*\ln(x))$.

Примеры записи арифметических выражений

Математическая запись	Запись на школьном алгоритмическом языке
$\frac{xy}{z}$	<code>x * y / z</code>
$\frac{x}{yz}$	<code>x / (y * z)</code> или <code>x / y / z</code>
$\frac{a^3 + b^3}{bc}$	<code>(a**3 + b**3) / (b*c)</code>
$\frac{a_{i+1} + b_{i-1}}{2xy}$	<code>(a[i+1] + b[i-1]) / (2*x*y)</code>
$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$	<code>(-b + sqrt(b*b - 4*a*c)) / (2*a)</code>
$\sqrt[5]{x} (x < 0)$	<code>sign(x) * abs(x) ** (1/5)</code>
$0,49e^{a^b - b^b} + \ln^3 \cos a^2$	<code>0.49 * exp(a*a - b*b) + ln(cos(a*a)) ** 3</code>
$\frac{x}{1 + \frac{x^2}{3 + (2x)^3}}$	<code>x / (1 + x*x / (3 + (2*x)**3))</code>

Типичные ошибки в записи выражений:

$5x + 1$
 $a + \sin x$ Пропущен знак умножения между 5 и x
 x Аргумент x функции $\sin x$ не заключен в скобки
 $(a + b) / c^{**3}$ Не хватает закрывающей скобки

7.21. Как записываются логические выражения?

В записи логических выражений помимо арифметических операций сложения, вычитания, умножения, деления и возведения в степень **используются операции отношения** $<$ (меньше), $<=$ (меньше или равно), $>$ (больше), $>=$ (больше или равно), $=$ (равно), $<>$ (не равно), **а также логические операции и, или, не.**

Примеры записи логических выражений, истинных при выполнении указанных условий.

Условие	Запись на школьном алгоритмическом языке
Дробная часть вещественного числа a равна нулю	$\text{int}(a) = 0$
Целое число a — четное	$\text{mod}(a, 2) = 0$
Целое число a — нечетное	$\text{mod}(a, 2) = 1$
Целое число k кратно семи	$\text{mod}(a, 7) = 0$
Каждое из чисел a, b положительно	$(a > 0)$ и $(b > 0)$
Только одно из чисел a, b положительно	$((a > 0)$ и $(b \leq 0))$ или $((a \leq 0)$ и $(b > 0))$
Хотя бы одно из чисел a, b, c является отрицательным	$(a < 0)$ или $(b < 0)$ или $(c < 0)$
Число x удовлетворяет условию $a < x < b$	$(x > a)$ и $(x < b)$
Число x имеет значение в промежутке [1, 3]	$(x \geq 1)$ и $(x \leq 3)$
Целые числа a и b имеют одинаковую четность	$((\text{mod}(a, 2) = 0)$ и $(\text{mod}(b, 2) = 0))$ или $((\text{mod}(a, 2) = 1)$ и $(\text{mod}(b, 2) = 1))$
Точка с координатами (x, y) лежит в круге радиуса r с центром в точке (a, b)	$(x - a)^2 + (y - b)^2 < r^2$
Уравнение $ax^2 + bx + c = 0$ не имеет действительных корней	$b^2 - 4ac < 0$
Точка (x, y) принадлежит первой или третьей четверти	$((x > 0)$ и $(y > 0))$ или $((x < 0)$ и $(y < 0))$
Точка (x, y) принадлежит внешности единичного круга с центром в начале координат или его второй четверти	$(x^2 + y^2 > 1)$ или $((x^2 + y^2 \leq 1)$ и $(x < 0)$ и $(y > 0))$
Целые числа a и b являются взаимнопротивоположными	$a = -b$
Целые числа a и b являются взаимнообратными	$a * b = 1$
Число a больше среднего арифметического чисел b, c, d	$a > (b + c + d) / 3$
Число a не меньше среднего геометрического чисел b, c, d	$a \geq (b * c * d)^{(1/3)}$
Хотя бы одна из логических переменных $F1$ и $F2$ имеет значение да	$F1$ или $F2$
Обе логические переменные $F1$ и $F2$ имеют значение да	$F1$ и $F2$
Обе логические переменные $F1$ и $F2$ имеют значение нет	не $F1$ и не $F2$

Логическая переменная F1 имеет значение да , а логическая переменная F2 имеет значение нет	F1 и не F2
Только одна из логических переменных F1 и F2 имеет значение да	(F1 и не F2) или (F2 и не F1)

7.22. Упражнения

7.1. Запишите по правилам алгоритмического языка выражения:

а) $\frac{x+y}{x-1/2} - \frac{x-z}{xy}$;

е) $\frac{\sqrt{|\sin^2 x|}}{3,01x - e^{2x}}$;

б) $(1+z) \frac{x + \frac{y}{z}}{a - \frac{1}{1+x^y}}$;

ж) $\frac{|\cos x^2 - \sin^2 y|}{\sqrt[4]{|\ln x| + xy}}$;

в) $(x^n)^{m+2} + x^{n^m}$;

з) $\ln(y^{-\sqrt{x+1}}) \cdot \sin^2 \operatorname{arctg} z$

г) $\frac{(a+b)^n}{1 + \frac{a}{a^m - b^{m-n}}}$;

и) $r_{ij}^{|x-y|} - 0,15 |\sin e^{-z^2}|$;

д) $\frac{(a_i^{2l} + b_{j+1}^{2k}) \cdot (3^n - x^2 y)}{z - \frac{d_{i,j+1+1}}{z + \frac{y}{\sqrt{t^2 + xyz}}}}$;

к) $a^{(x+y)/2} - 3 \sqrt[3]{\frac{x-1}{|y|+1}} \cdot e^{-(y+u/2)}$

7.2. Запишите в обычной математической форме арифметические выражения:

а) $a / b ** 2$;

л) $5 * \operatorname{arctg}(x) - \operatorname{arctg}(y) / 4$;

б) $a+b/c+1$;

м) $\lg(u * (1/3) + \operatorname{sqrt}(v) + z)$;

в) $1/a*b/c$;

н) $\ln(y * (-\operatorname{sqrt}(\operatorname{abs}(x))))$;

г) $a**b**c/2$;

о) $\operatorname{abs}(x**(y/x) - (y/x)**(1/3))$;

д) $(a**b)**c/2$;

п) $\operatorname{sqrt}((x1-x2)**2 + (y1-y2)**2)$;

е) $a/b/c/d*p*q$;

р) $\exp(\operatorname{abs}(x-y)) * (\operatorname{tg}(z)**2+1)**x$;

ж) $x**y**z/a/b$;

с) $\lg(\operatorname{sqrt}(\exp(x-y)) + x**\operatorname{abs}(y) + z)$;

з) $4/3*3.14*r**3$;

т) $\operatorname{sqrt}(\exp(a*x) * \sin(x)**n / \cos(x)**2)$;

и) $b/\operatorname{sqrt}(a*a+b)$;

у) $\operatorname{sqrt}(\sin(\operatorname{arctg}(u))**2 + \operatorname{abs}(\cos(v)))$;

к) $d*c/2/R+a**3$;

ф) $\operatorname{abs}(\cos(x) + \cos(y))** (1 + \sin(y)**2)$;

7.3. Вычислите значения арифметических выражений при $x=1$:

а) $\operatorname{abs}(x-3) / \ln(\exp(3)) * 2 / \lg(10000)$;

Решение: $\operatorname{abs}(1-3)=2$; $\ln(\exp(3))=3$; $\lg(10000)=4$; $2/3*2/4=0.33$;

б) $\operatorname{sign}(\operatorname{sqrt}(\operatorname{sqrt}(x+15))) * 2**2**2$;

в) $\operatorname{int}(-2.1) * \operatorname{int}(-2.9) / \operatorname{int}(2.9) + x$;

г) $-\operatorname{sqrt}(x+3)**2**(\operatorname{sign}(x+0.5)*3) + \operatorname{tg}(0)$;

- д) $\lg(x) + \cos(x^2 - 1) * \sqrt{x + 8} - \text{div}(2, 5)$;
 е) $\text{sign}(x - 2) * \sqrt{\text{int}(4.3)} / \text{abs}(\text{min}(2, -1))$;
 ж) $\text{div}(10, x + 2) * \text{mod}(10, x + 6) / \text{max}(10, x) * \text{mod}(2, 5)$.

7.4. Запишите арифметические выражения, значениями которых являются:

- а) площадь треугольника со сторонами a, b, c ($a, b, c > 0$) и полупериметром p ;

Ответ: $\sqrt{p * (p - a) * (p - b) * (p - c)}$;

- б) среднее арифметическое и среднее геометрическое чисел a, b, c, d ;
 в) расстояние от точки с координатами (x, y) до точки $(0, 0)$;
 г) синус от x градусов;
 д) площадь поверхности куба (длина ребра равна a);
 е) радиус описанной сферы куба (длина ребра равна a);
 ж) координаты точки пересечения двух прямых, заданных уравнениями
 $a_1x + b_1y + c_1 = 0$ и $a_2x + b_2y + c_2 = 0$ (прямые не параллельны).

7.5. Вычислите значения логических выражений:

- а) $x * x + y * y <= 9$ при $x = 1, y = -2$

Ответ: да;

- б) $b * b - 4 * a * c < 0$ при $a = 2, b = 1, c = -2$;
 в) $(a >= 1)$ **и** $(a <= 2)$ при $a = 1.5$;
 г) $(a < 1)$ **или** $(a > 1.2)$ при $a = 1.5$;
 д) $(\text{mod}(a, 7) = 1)$ **и** $(\text{div}(a, 7) = 1)$ при $a = 8$;
 е) **не** $((a > b)$ **и** $(a < 9)$ **или** $(a * a = 4))$ при $a = 5, b = 4$.

7.6. Запишите логические выражения, истинные только при выполнении указанных условий:

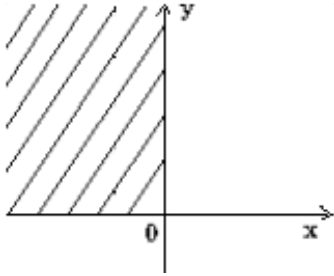
- а) x принадлежит отрезку $[a, b]$

Ответ: $(x >= a)$ **и** $(x <= b)$;

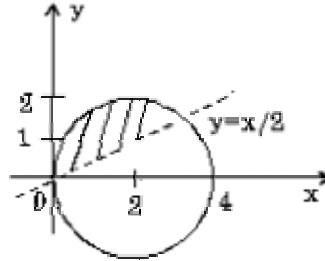
- б) x лежит вне отрезка $[a, b]$;
 в) x принадлежит отрезку $[a, b]$ или отрезку $[c, d]$;
 г) x лежит вне отрезков $[a, b]$ и $[c, d]$;
 д) целое k является нечетным числом;
 е) целое k является трехзначным числом, кратным пяти;
 ж) элемент $a_{i,j}$ двумерного массива находится на пересечении нечетной строки и четного столбца;
 з) прямые $a_1x + b_1y + c_1 = 0$ и $a_2x + b_2y + c_2 = 0$ параллельны;
 и) из чисел a, b, c меньшим является c , а большим b ;
 к) среди чисел a, b, c, d есть взаимно противоположные;
 л) среди целых чисел a, b, c есть хотя бы два четных;
 м) из отрезков с длинами a, b, c можно построить треугольник;
 н) треугольники со сторонами a_1, b_1, c_1 и a_2, b_2, c_2 подобны;
 о) точка с координатами (x, y) принадлежит внутренней области треугольника с вершинами $A(0, 5), B(5, 0)$ и $C(1, 0)$;
 п) точка с координатами (x, y) принадлежит области, внешней по отношению к треугольнику с вершинами $A(0, 5), B(1, 0)$ и $C(5, 0)$;
 р) четырехугольник со сторонами a, b, c и d является ромбом.

7.7. Начертите на плоскости (x,y) область, в которой и только в которой истинно указанное выражение. Границу, не принадлежащую этой области, изобразите пунктиром.

а) $(x \leq 0)$ и $(y \geq 0)$
Ответ:

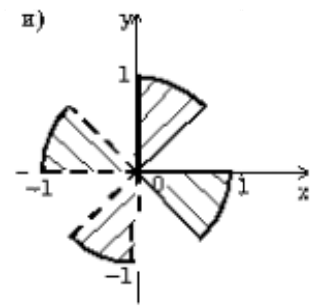
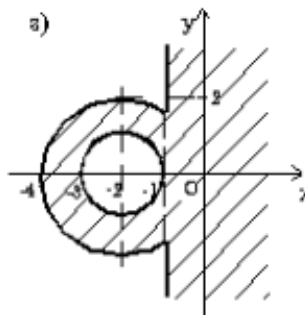
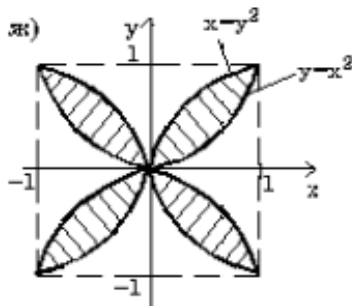
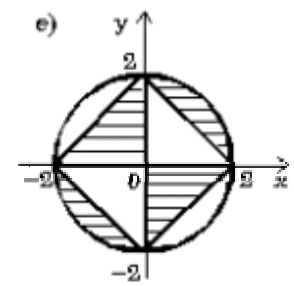
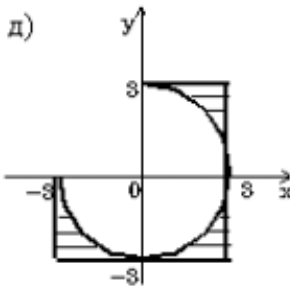
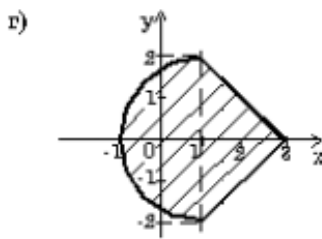
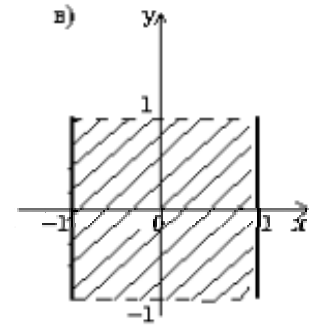
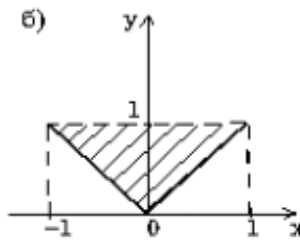
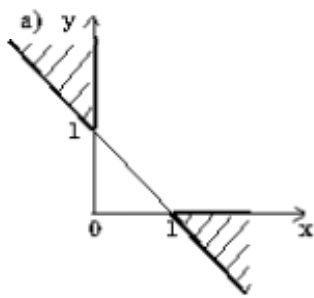


е) $((x-2)^2 + y^2 \leq 4)$ и $(y > x/2)$
Ответ:



б) $(x \geq 0)$ или $(y \leq 0)$ **ж)** $(x^2 + y^2 < 1)$ и $(y > x^2)$;
в) $x + y \geq 0$ **з)** $(y > x)$ и $(y + x \geq 0)$ и $(y \leq 1)$;
г) $(x + y > 0)$ и $(y < 0)$ **и)** $(\text{abs}(x) \leq 1)$ и $(y < 2)$;
д) $\text{abs}(x) + \text{abs}(y) \geq 1$ **к)** $(x^2 + y^2 < 4)$ и $(x^2 + y^2 > 1)$;

7.8. Запишите логическое выражение, которое принимает значение "истина" тогда и только тогда, когда точка с координатами (x, y) принадлежит заштрихованной области.



7.9. Пусть $a=3$, $b=5$, $c=7$. Какие значения будут иметь эти переменные в результате выполнения последовательности операторов:

а) $a:=a+1$; $b:=a+b$; $c:=a+b$; $a:=\text{sqrt}(a)$

Решение: $a=3+1=4$, $b=4+5=9$, $c=4+9=13$, $a = \{\text{корень квадратный из}\} 4 = 2$.

Ответ: $a=2$, $b=9$, $c=13$;

б) $c:=a*b+2$; $b:=b+1$; $a:=c-b**2$; $b:=b*a$;

в) $b:=b+a$; $c:=c+b$; $b:=1/b*c$;

г) $p:=c$; $c:=b$; $b:=a$; $a:=p$; $c:=a*b*c*p$;

д) $c:=a**(b-3)$; $b:=b-3$; $a:=(c+1)/2*b$; $c:=(a+b)*a$;

е) $x:=a$; $a:=b$; $b:=c$; $c:=x$; $a:=\text{sqrt}(a+b+c+x-2)$;

ж) $b:=(a+c)**2$; $a:=\lg(b**2)**2$; $c:=c*a*b$.

7.10. Задайте с помощью операторов присваивания следующие действия:

а) массив $X=(x_1, x_2)$ преобразовать по правилу: в качестве x_1 взять сумму, а в качестве x_2 — произведение исходных компонент;

Решение: $c:=x[1]$; $x[1]:=x[1]+x[2]$; $x[2]:=c*x[2]$

б) поменять местами значения элементов массива $X=(x_1, x_2)$;

в) в массиве $A(N)$ компоненту с номером i ($1 < i < N$) заменить полусуммой исходных соседних с нею компонент, соседнюю справа компоненту заменить на нуль, а соседнюю слева компоненту увеличить на 0.5;

г) $u = \max(x, y, z) + \min(x-z, y+z, y, z)$;

7.11. Задайте с помощью команд **если** или **выбор** вычисления по формулам:

$$\text{а) } y = \begin{cases} \sqrt[3]{x} & , \text{ если } x \leq -100, \\ \sqrt[3]{x} & , \text{ если } -100 < x < 100, \\ \sqrt{x} & , \text{ если } x \geq 100; \end{cases}$$

$$\text{б) } z = \begin{cases} x^2 + y^2 & , \text{ если } x^2 + y^2 \leq 1, \\ x + y & , \text{ если } x^2 + y^2 > 1 \text{ и } y \geq x, \\ 0,5 & , \text{ если } x^2 + y^2 > 1 \text{ и } y < x; \end{cases}$$

$$F(z) = \begin{cases} 2z + 1, & \text{ если } z \geq 0, \\ \sin z, & \text{ если } z < 0, \end{cases}$$

в) где

$$z = \begin{cases} \lg(-x), & \text{ если } x < 0, \\ \sqrt{x+1}, & \text{ если } x \geq 0. \end{cases}$$

$$\text{г) } z = \begin{cases} 1 & , \text{ если } c = 0, \\ x & , \text{ если } c = 1, \\ 3x^2 - 1/2 & , \text{ если } c = 2, \\ x^3 - 3x/2 & , \text{ если } c = 3, \\ 2x^4 - 3x/2 & , \text{ в противном} \\ & \text{случае} \end{cases}$$

$$д) z = \begin{cases} \sqrt{x^2 + y^2} & , \text{ если } |x| + |y| < r, \\ \max(|x|, |y|) & , \text{ если } |x| + |y| \geq r, \end{cases}$$

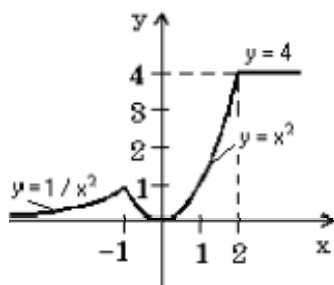
$$е) v = \begin{cases} x + y & , \text{ если } x > 1 \text{ и } y > 1, \\ x - y & , \text{ если } x > 1 \text{ и } y \leq 1, \\ -x + y & , \text{ если } x \leq 1 \text{ и } y > 0, \\ -x - y & , \text{ если } x \leq 1 \text{ и } y \leq 0. \end{cases}$$

$$ж) z = \begin{cases} |x| + |y| & , \text{ если точка лежит внутри круга радиусом } r (r > 0) \text{ с центром в} \\ & \text{точке} \\ & (a, b) \\ x + y & , \text{ в противном случае} \end{cases}$$

7.12. Постройте графики функций $y(x)$, заданных командами **если**:

```
а)  если x <= -1
    то y := 1/x**2
    иначе
      если x <= 2
      то y := x*x
      иначе y := 4
    все
все
```

```
в)  если x < -0.5
    то y := 1/abs(x)
    иначе
      если x < 1
      то y := 2
      иначе y := 1/(x-0.5)
    все
все
```



Решение

```
г)  если x < 0
    то y := 1
    иначе
      если x < 3.14
      то y := cos(x)
      иначе y := -1
    все
все
```

```
б)  если x < -5
    то y := -5
    иначе
      если x < 0
      то y := x
      иначе
        если x < 3
        то y := 2*x
        иначе y := 6
      все
    все
все
```

```
д)  если abs(x) > 2
    то y := x*x
    иначе
      если x < 0
      то y := -2*x
      иначе
        если x >= 1
        то y := 4
        иначе y := 4*x*x
      все
    все
все
```


7.13. Определите значение целочисленной переменной S после выполнения операторов:

а) S:=128
нц для i от 1 до
 4
 S:=div(S,2)
кц

Решение

i	S
	128
1	128/2=64
2	64/2=32
3	32/2=16
4	16/2=8

Ответ: S=8

б) S:=1; a:=1
нц для i от 1 до 3
 S:=S+i*(i+1)*a
 a:=a+2
кц

в) S:=1; a:=1
нц для i от 1 до 3
 S := S+i
 нц для j от 2 до 3
 S := S+j
 кц
кц

7.14. Определите значение переменной S после выполнения операторов:

а) i:=0; S:=0
нц пока i < 3
 i:=i+1;
 S:=S+i*i
кц

Решение

Условие $i < 3$	i	S
	0	0
$0 < 3?$ да	1	$0+1^2=1$
$1 < 3?$ да	2	$1+2^2=5$
$2 < 3?$ да	3	$5+3^2=14$
$3 < 3?$ нет(кц)		

Ответ: S=14

б) S:=0; i:=1

г) S:=0
нц для i от 1 до
 2
 нц для j от 2
 до 3
 S:=S+i+j
 кц
кц

Решение

i	j	S
		0
1	2	$0+1+2=3$
	3	$3+1+3=7$
2	2	$7+2+2=11$
	3	$11+2+3=16$

Ответ: S=16

д) **нц для i от 1 до 3**
 S:=0
 нц для j от 2 до 3
 S:=S+i+j
 кц
кц

е) **нц для i от 1 до 2**
 S := 0
 нц для j от 2 до 3
 нц для k от 1 до 2
 S := S+i+j+k
 кц
 кц
кц

г) S:=0; N:=125
нц пока N > 0
 S:=S+mod(N,10) | S – сумма
 цифр
 N:=div(N,10) | числа N
кц

Решение

Условие $N > 0$	S	N
	0	125
$125 > 0?$ да	$0+5=5$	12
$12 > 0?$ да	$5+2=7$	1
$1 > 0?$ да	$7+1=8$	0
$0 > 0?$ нет(кц)		

Ответ: S=8

д) a:=1; b:=1; S:=0;

```

нц пока i>1
  S:=S+1/i
  i:=i-1
кц

```

```

нц пока a<=5
  a:=a+b; b:=b+a;
  S:=S+a+b
кц

```

```

в) S:=0; i:=1; j:=5
нц пока i<j
  S:=S+i*j
  i:=i+1
  j:=j-1
кц

```

```

е) a:=1; b:=1
нц пока a+b<10
  a:=a+1
  b:=b+a
кц
  S:=a+b

```

7.15. Составьте алгоритмы решения задач линейной структуры (условия этих задач заимствованы из учебного пособия В.М. Заварыкина, В.Г. Житомирского и М.П. Лапчика "Основы информатики и вычислительной техники", 1989):

а) в треугольнике известны три стороны a , b и c ; найти (в градусах) углы этого треугольника, используя формулы:

$$\cos A = \frac{b^2 + c^2 - a^2}{2bc}; \quad \sin B = \frac{b \sin A}{a}; \quad C = 180^\circ - (A + B).$$

Пояснение. Обратите внимание на то, что стандартные тригонометрические функции \arccos и \arcsin возвращают вычисленное значение в радианной мере.

Решение:

```

алг Углы треугольника (арг вещь a, b, c, рез вещь UголA, UголB, UголC)
нач вещь RadGr, UголARad
  | RadGr — коэф. перевода угла из радианной меры в градусную
  | UголARad — угол A (в радианах)
  RadGr:=180/3.14
  UголARad:=ArcCos((b*b+c*c-a*a)/(2*b*c))
  UголA:=UголARad*RadGr
  UголB:=ArcSin(b*sin(UголARad)/a)*RadGr
  UголC:=180-(UголA+UголB)
кон

```

б) в треугольнике известны две стороны a , b и угол C (в радианах) между ними; найти сторону c , углы A и B (в радианах) и площадь треугольника, используя формулы:

$$\sin A = \frac{a \sin C}{c}; \quad \sin B = \frac{b \sin C}{c};$$

$$S = \frac{bc \sin A}{2}; \quad c^2 = a^2 + b^2 - 2ab \cos C.$$

Пояснение. Сначала нужно найти сторону c , а затем остальные требуемые значения;

в) в треугольнике известны три стороны a , b и c ; найти радиус описанной окружности и угол A (в градусах), используя формулы:

$$S = \frac{bc \sin A}{2}; \quad \operatorname{tg} \frac{A}{2} = \sqrt{\frac{(p-b)(p-c)}{p(p-a)}}, \quad \text{где } p = \frac{a+b+c}{2}.$$

г) в правильной треугольной пирамиде известны сторона основания a и угол A (в градусах) наклона боковой грани к плоскости основания; найти объем и площадь полной поверхности пирамиды, используя формулы:

$$V = S_{\text{осн}} \cdot H/3; \quad S_{\text{п}} = S_{\text{осн}} \cdot \left(1 + \frac{1}{\cos \alpha}\right),$$

где $S_{\text{осн}} = \frac{a^2 \sqrt{3}}{4}$, $H = \frac{a \sqrt{3}}{6} \operatorname{tg} \alpha$.

д) в усеченном конусе известны радиусы оснований R и r и угол A (в радианах) наклона образующей к поверхности большего основания; найти объем и площадь боковой поверхности конуса, используя формулы:

$$V = \frac{1}{3} \pi H (r^2 + R^2 + rR); \quad S_{\text{бок}} = \pi l (r+R)$$

где $H = (R-r) \operatorname{tg} \alpha$ $l = \frac{R-r}{\cos \alpha}$.

е) в правильной четырехугольной пирамиде сторона основания равна a , а боковое ребро наклонено к плоскости основания под углом A ; найти объем и площадь полной поверхности пирамиды и площадь сечения, проходящего через вершину пирамиды и диагональ основания d ; использовать формулы:

$$H = \frac{a \sqrt{2}}{2} \cdot \operatorname{tg} \alpha; \quad V = \frac{1}{3} \cdot S_{\text{осн}} \cdot H;$$

$$S_{\text{сеч}} = \frac{1}{2} H a \sqrt{2} = \frac{a H \sqrt{2}}{2}; \quad S_{\text{полн}} = a^2 + a^2 \sqrt{2} \operatorname{tg}^2 \alpha + 1.$$

7.16. Составьте алгоритм решения задач разветвляющейся структуры:

а) определить, является ли треугольник с заданными сторонами a , b , c равнобедренным;

Решение:

```

алг Треугольник (арг вещь a, b, c, рез лог Otvet)
  дано | a > 0, b > 0, c > 0, a + b > c, a + c > b, b + c > a
  надо | Otvet = да, если треугольник равнобедренный
        | Otvet = нет, если треугольник не равнобедренный
нач
  если (a = b) или (a = c) или (b = c)
  то Otvet := да
  иначе Otvet := нет
все
кон
  
```

б) определить количество положительных чисел среди заданных чисел a , b и c ;

в) меньшее из двух заданных неравных чисел увеличить вдвое, а большее оставить без изменения;

г) числа a и b — катеты одного прямоугольного треугольника, а c и d — другого; определить, являются ли эти треугольники подобными;

д) даны три точки на плоскости; определить, какая из них ближе к началу координат;

е) определить, принадлежит ли заданная точка (x, y) плоской фигуре, являющейся кольцом с центром в начале координат, с внутренним радиусом r_1 и внешним радиусом r_2 ;

ж) упорядочить по возрастанию последовательность трех чисел a , b и c .

Ответы — Раздел 7. Алгоритмы. Алгоритмизация. Алгоритмические языки

7.1.

- а) $(x+y)/(x-1/2) - (x-z)/(x*y)$;
 б) $(1+z) * (x+y/z) / (a-1/(1+x*x))$;
 в) $x**(n*(m+2)) + x**(n**m)$;
 г) $(a+b)**n / (1+a/(a**m-b**(m-n)))$;
 д) $(a[i]**2 + b[j+1]**2) * (3**n-x*x*y) / (z - (d[i, j+1]+1) / (z + y/sqrt(t*t+x*y*z)))$;
 е) $sqrt(abs(sin(x)**2)) / (3.01*x - exp(2*x))$;
 ж) $abs(cos(x**3) - sin(y)**2) / (abs(ln(x))**(1/4) + x*y)$;
 з) $ln(y**(-sqrt(abs(x+1)))) * sin(arctg(z))**2$;
 и) $r[i, j]**abs(x-y) - 0.15*abs(sin(exp(-z**8)))$;
 к) $a**((x+y)/2) - ((x-1)/(abs(y)+1))**(1/3)*exp(-(y+u/2))$.

- 7.2. а) $\frac{a}{b^2}$; б) $a + \frac{b}{c} + 1$; в) $\frac{b}{a \cdot c}$; г) $\frac{a^{b^c}}{2}$; д) $\frac{a^{bc}}{2}$; е) $\frac{a \cdot p \cdot q}{b \cdot c \cdot d}$; ж) $\frac{x^{y^z}}{a \cdot b}$;
 з) $\frac{4\pi^3}{3}$; и) $\frac{b}{\sqrt{a^2 + b}}$; к) $\frac{dc}{2R} + a^3$; л) $\frac{\arctg y}{4}$; м) $\lg\left(\frac{u}{3} + \sqrt{v+z}\right)$;
 н) $\ln(-y\sqrt{|x|})$; о) $\left| x^{y^k} - 3\sqrt{\frac{y}{x}} \right|$; п) $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$; р) $e^{k-y} \cdot (tg^2 z + 1)^x$;
 с) $\lg\left(\sqrt{e^{x-y}} + x^{|y|} + z\right)$; т) $\frac{\sqrt{e^{ax} \cdot \sin^n x}}{\cos^2 x}$; у) $\sqrt{\sin^2 \arctg u + |\cos v|}$;
 ф) $|\cos x + \cos y|^{1+\sin^2 y}$.

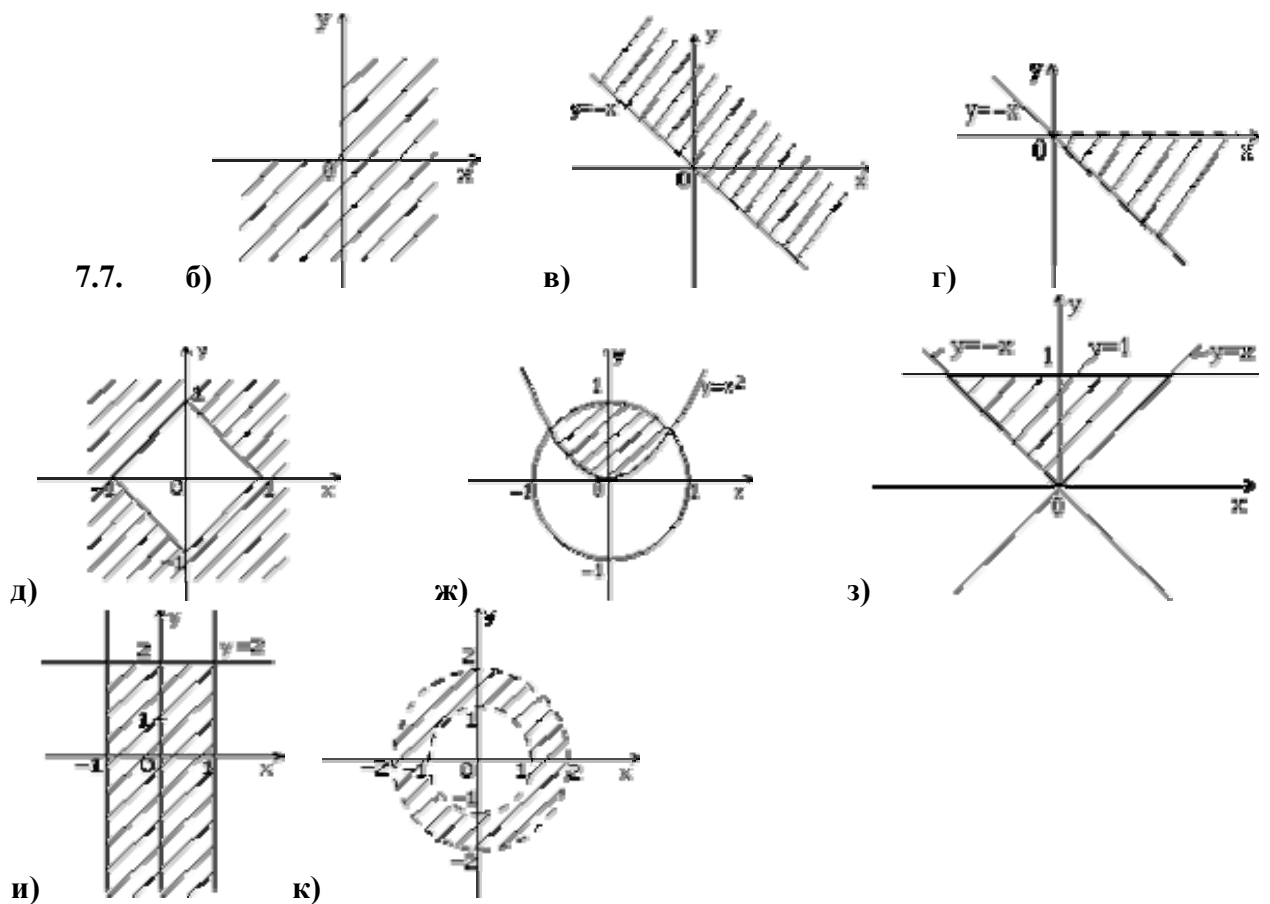
7.3. б) 16; в) 5,5; г) -256; д) 3; е) -2; ж) 1.8.

7.4. б) среднее арифметическое: $(a+b+c+d)/4$; среднее геометрическое: $(a*b*c*d)^{(1/4)}$; в) $\sqrt{x*x+y*y}$; г) $\sin(x*3.14/180)$; д) $6*a*a$; е) $\sqrt{3}*a/2$; ж) абсцисса: $(c1*b2-c2*b1)/(b1*a2-b2*a1)$; ордината: $(c2*a1-c1*a2)/(b1*a2-b2*a1)$.

7.5. б) нет; в) да; г) да; д) да; е) нет;

7.6.

- б) $(x < a)$ или $(x > b)$;
в) $((x \geq a) \text{ и } (x \leq b))$ или $((x \geq c) \text{ и } (x \leq d))$;
г) $((x < a) \text{ или } (x > b))$ и $((x < c) \text{ или } (x > d))$;
д) $\text{mod}(k, 2) = 1$;
е) $(\text{mod}(k, 5) = 0)$ и $(k > 99)$ и $(k < 1000)$;
ж) $(\text{mod}(i, 2) = 1)$ и $(\text{mod}(j, 2) = 0)$;
з) $a1*b2 = a2*b1$;
и) $(c < a)$ и $(b > a)$;
к) $(a = -b)$ или $(a = -c)$ или $(a = -d)$ или $(b = -c)$ или $(b = -d)$ или $(c = -d)$;
л) $((\text{mod}(a, 2) = 0) \text{ и } (\text{mod}(b, 2) = 0))$ или $((\text{mod}(a, 2) = 0) \text{ и } (\text{mod}(c, 2) = 0))$ или $((\text{mod}(b, 2) = 0) \text{ и } (\text{mod}(c, 2) = 0))$;
м) $(a > 0)$ и $(b > 0)$ и $(c > 0)$ и $(a+b > c)$ и $(a+c > b)$ и $(b+c > a)$;
н) $((a1*b2 = a2*b1) \text{ и } (a1*c2 = a2*c1))$ или $((a1*c2 = a2*b1) \text{ и } (a1*b2 = a2*c1))$ или $((a1*c2 = b2*b1) \text{ и } (a1*a2 = b2*c1))$ или $((a1*a2 = b2*b1) \text{ и } (a1*c2 = b2*c1))$ или $((a1*a2 = c2*b1) \text{ и } (a1*b2 = c2*c1))$ или $((a1*b2 = c2*b1) \text{ и } (a1*a2 = c2*c1))$;
о) $(y > 5 - 5*x)$ и $(y < 5 - x)$ и $(y > 0)$;
п) $(y < 5 - 5*x)$ или $(y > 5 - x)$ или $(y < 0)$;
р) $(a = b)$ и $(c = d)$ и $(b = c)$.
-



7.8.

- а) $(y \geq 1 - x)$ и $((y \leq 0)$ или $(x \leq 0))$;
 б) $(y < 1)$ и $(y \geq x)$ и $(y \geq -x)$ (вариант ответа: $(y < 1)$ и $(y \geq \text{abs}(x))$);
 в) $(\text{abs}(x) \leq 1)$ и $(\text{abs}(y) < 1)$;
 г) $((x - 1) * (x - 1) + y * y \leq 4)$ и $(y \leq 3 - x)$ и $(y \geq x - 3)$;
 д) $(\text{abs}(x) \leq 3)$ и $(\text{abs}(y) \leq 3)$ и $(x * x + y * y \geq 9)$ и $((x \geq 0)$ или $(y \leq 0))$;
 е) $(\text{abs}(x) + \text{abs}(y) \leq 2)$ и $(\text{sign}(x) \neq \text{sign}(y))$ или $(x * x + y * y \leq 4)$
 и $((y \geq 2 - x)$ или $(y \leq -x - 2))$;
 ж) $((y \geq x * x)$ или $(y \leq -x * x))$ и $((x \geq y * y)$ или $(x \leq -y * y))$;
 з) $((x + 2) * (x + 2) + y * y \leq 4)$ и $((x + 2) * (x + 2) + y * y \geq 1)$ или $(x \geq -1)$;
 и) $((y \leq 0) = (y \geq -x))$ или $((x \geq 0) = (y \geq x))$ и $(x * x + y * y \leq 1)$.

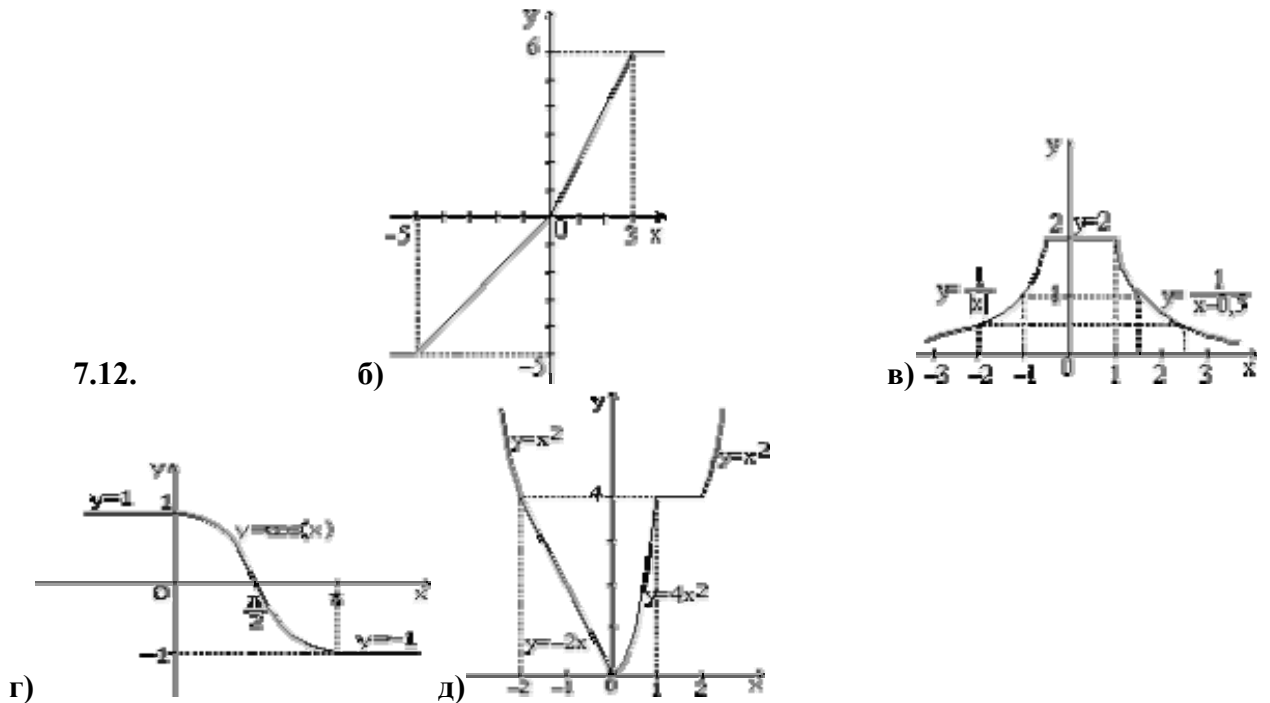
7.9. б) $a = -19$; $b = -114$; $c = 17$; в) $a = 3$; $b = 1,875$; $c = 15$; г) $a = 7$; $b = 3$;
 $c = 735$; д) $a = 10$; $b = 2$; $c = 120$; е) $a = 4$; $b = 7$; $c = 3$; ж) $a = 16$; $b = 100$,
 $c = 11200$.

7.10. б) $c := x[1]$; $x[1] := x[2]$; $x[2] := c$; в) $a[i] := (a[i-1] + a[i+1]) / 2$; $a[i+1] := 0$; $a[i-1] := a[i-1] + 0.5$; г) $u := \max(\max(x, y), z) + \min(\min(x - z, y + z), \min(y, z))$.

7.11.

- а) **если** $x \leq -100$
то $y := \text{sign}(x) * \text{abs}(x) ** (1/7)$
иначе если $x < 100$
то $y := \text{sign}(x) * \text{abs}(x) ** (1/3)$
иначе $y := \text{sqrt}(x)$
все
все
- б) **если** $x*x+y*y \leq 1$
то $z := x*x+y*y$
иначе если $y > x$
то $z := x+y$
иначе $z := 0.5$
все
все
- в) **если** $x < 0$
то $z := \lg(-x)$
иначе $z := \text{sqrt}(x+1)$
все
если $z >= 0$
то $F := 2*z+1$
иначе $F := \sin(z)$
все
- г) **выбор**
при $c=0$: $z:=1$
при $c=1$: $z:=x$
при $c=2$: $z:=3*x*x - 1/2$
при $c=3$: $z:=x*x*x - 3*x/2$
иначе $z:=2*x**4 - 3*x/2$
все
- д) **если** $\text{abs}(x)+\text{abs}(y) < r$
то $z := \text{sqrt}(x*x+y*y)$
иначе $z := \max(\text{abs}(x), \text{abs}(y))$
все
- е) **если** $x > 1$
то если $y > 1$
то $v := x+y$
иначе $v := x-y$
все
иначе если $y > 0$
то $v := y-x$
иначе $v := -x-y$
все
все
- ж) **если** $(x-a)**2 + (y-b)**2 < r*r$
то $z := \text{abs}(x)+\text{abs}(y)$
иначе $z := x+y$
все

7.12.



7.13. б) 81; в) 21; д) 11; е) 44.

7.14. б) 0; в) 13; д) 52; е) 14.

7.15.

- б) **алг** Треугольник1 (**арг вещь** a,b,UgolC, **рез вещь** c, UgolA, UgolB, S)
нач
 ввод a, b, UgolC
 $c := \sqrt{a^2 + b^2 - 2 * a * b * \cos(UgolC)}$
 $UgolA := \arcsin(a * \sin(UgolC) / c)$
 $UgolB := \arcsin(b * \sin(UgolC) / c)$
 $S := b * c * \sin(UgolA) / 2$
 вывод c, UgolA, UgolB, S
кон
- в) **алг** Треугольник2 (**арг вещь** a,b,c, **рез вещь** Radius,UgolA)
нач вещь p
 ввод a,b,c
 $p := (a + b + c) / 2$
 $UgolA := 2 * \arctg(\sqrt{(p - b) * (p - c) / (p * (p - a))}) * 180 / 3.14$
 $Radius := a * b * c / (4 * \sqrt{p * (p - a) * (p - b) * (p - c)})$
 вывод Radius, UgolA
кон
- г) **алг** Объем и Площадь Пирамиды (**арг вещь** a,UgolAGrad, **рез вещь** V, S)
нач вещь H, SBase, UgolARad
 | H - высота пирамиды; SBase - площадь основания
 ввод a,UgolAGrad
 $UgolARad := UgolAGrad * 3.14 / 180$
 $SBase := a * a * \sqrt{3} / 4$
 $H := a * \sqrt{3} / 6 * \tg(UgolARad)$
 $V := SBase * H / 3$
 $S := SBase * (1 + 1 / \cos(UgolARad))$
 вывод V, S
кон
- д) **алг** Объем и Площадь конуса (**арг вещь** RBig,RSmall,Ugol, **рез вещь** V, S)
нач вещь H,L
 ввод RBig,RSmall,Ugol
 $H := (RBig - RSmall) * \tg(Ugol)$
 $L := (RBig - RSmall) / \cos(Ugol)$
 $V := 1 / 3 * 3.14 * H * (RSmall^2 + RBig^2 + RSmall * RBig)$
 $S := 3.14 * L * (RBig + RSmall)$
 вывод V, S
кон
- е) **алг** Параметры пирамиды (**арг вещь** a,UgolA, **рез вещь** V, S, Section)
нач вещь H
 ввод a,UgolA
 $H := a * \sqrt{2} / 2 * \tg(UgolA)$
 $V := 1 / 3 * a * a * H$
 $Section := a * H * \sqrt{2} / 2$
 $S := a * a * (1 + \sqrt{2 * \tg(UgolA)^2 + 1})$
 вывод V, S, Section
кон
-

7.16.

- б) **алг** Количество положительных (**арг вещь** a,b,c, **рез** цел k)
 надо | k - количество положительных чисел среди чисел a,b,c
 нач
 ввод a,b,c; k:=0
 если a>0
 то k:=k+1
 все
 если b>0
 то k:=k+1
 все
 если c>0
 то k:=k+1
 все
 вывод k
 кон
- в) **алг** Преобразование (**арг рез вещь** a,b)
 надо | меньшее из a,b увеличено вдвое
 нач
 ввод a,b
 если a>b
 то b:=b*2
 иначе a:=a*2
 все
 вывод a,b
 кон
- г) **алг** Подобие треугольников (**арг вещь** a,b,c,d, **рез** лог Otvet)
 дано | a,b и c,d - катеты двух треугольников
 надо | Otvet=да, если треугольники подобны
 нач
 ввод a,b,c,d
 если (a*d=c*b) **или** (a*c=d*b)
 то Otvet:=да
 иначе Otvet:=нет
 все
 вывод Otvet
 кон
- д) **алг** Точки (**арг вещь** xA,yA,xB,yB,xC,yC, **рез** лит Otvet)
 нач **вещ** DistA,DistB,DistC
 ввод xA,yA,xB,yB,xC,yC
 DistA:=sqrt(xA**2 + yA**2)
 DistB:=sqrt(xB**2 + yB**2)
 DistC:=sqrt(xC**2 + yC**2)
 если (DistA < DistB) **и** (DistA < DistC)
 то Otvet:="Это точка А"
 иначе **если** DistB < DistC
 то Otvet:="Это точка В"
 иначе Otvet:="Это точка С"
 все
 все
 вывод Otvet
 кон
- е) **алг** Принадлежность кольцу (**арг вещь** x,y,r1,r2, **рез** лог Otvet)
 дано | r2>r1
 надо | Otvet=да, если точка (x,y) принадлежит кольцу
 | с внутренним радиусом r1 и внешним радиусом r2
 нач

```
ввод x, y, r1, r2
если (x*x+y*y<=r2*r2) и (x*x+y*y>=r1*r1)
  то Otvet:=да
  иначе Otvet:=нет
все
вывод Otvet
кон
```

```
ж) алг Упорядочение по возрастанию(арг рез вещь a, b, c)
  надо | числа a, b, c упорядочены по возрастанию
  нач вещь t
  ввод a, b, c
  если a>b
    то t:=a; a:=b; b:=t | меняются местами значения a и b
  все
  если a>c
    то t:=a; a:=c; c:=t | меняются местами значения a и c
  все
  если b>c
    то t:=b; b:=c; c:=t | меняются местами значения b и c
  все
  вывод a, b, c
кон
```

Глава 8. Технология подготовки и решения задач с помощью компьютера

8.1. Какие этапы включает в себя решение задач с помощью компьютера?

Решение задач с помощью компьютера включает в себя следующие основные этапы, часть из которых осуществляется без участия компьютера.

- 1. Постановка задачи:**
 - сбор информации о задаче;
 - формулировка условия задачи;
 - определение конечных целей решения задачи;
 - определение формы выдачи результатов;
 - описание данных (их типов, диапазонов величин, структуры и т.п.).
- 2. Анализ и исследование задачи, модели:**
 - анализ существующих аналогов;
 - анализ технических и программных средств;
 - разработка [математической модели](#);
 - разработка структур данных.
- 3. Разработка алгоритма:**
 - выбор метода проектирования алгоритма;
 - выбор формы записи алгоритма (блок-схемы, псевдокод и др.);
 - выбор [тестов](#) и метода тестирования;
 - проектирование алгоритма.
- 4. Программирование:**
 - выбор языка программирования;
 - уточнение способов организации данных;
 - запись алгоритма на выбранном языке программирования.
- 5. Тестирование и отладка:**
 - синтаксическая отладка;
 - отладка семантики и логической структуры;
 - тестовые расчеты и анализ результатов тестирования;
 - совершенствование программы.
- 6. Анализ результатов решения задачи** и уточнение в случае необходимости математической модели с повторным выполнением этапов 2 — 5.
- 7. Сопровождение программы:**
 - доработка программы для решения конкретных задач;
 - составление документации к решенной задаче, к математической модели, к алгоритму, к программе, к набору тестов, к использованию.

8.2. Что называют математической моделью?

Математическая модель — это система математических соотношений — формул, уравнений, неравенств и т.д., отражающих существенные свойства объекта или

явления.

Всякое явление природы бесконечно в своей сложности. Проиллюстрируем это с помощью примера, взятого из книги В.Н. Тростникова "Человек и информация" (Издательство "Наука", 1970).

... Обыватель формулирует математику задачу следующим образом: *"Сколько времени будет падать камень с высоты 200 метров?"* Математик начнет создавать свой вариант задачи приблизительно так: *"Будем считать, что камень падает в пустоте и что ускорение силы тяжести 9,8 метра в секунду за секунду. Тогда ..."*

— *Позвольте, — может сказать "заказчик", — меня не устраивает такое упрощение. Я хочу знать точно, сколько времени будет падать камень в реальных условиях, а не в несуществующей пустоте.*

— *Хорошо, — согласится математик. — Будем считать, что камень имеет сферическую форму и диаметр... Какого примерно он диаметра?*

— *Около пяти сантиметров. Но он вовсе не сферический, а продолговатый.*

— *Тогда будем считать, что он имеет форму эллипсоида с полуосями четыре, три и три сантиметра и что он падает так, что большая полуось все время остается вертикальной. Давление воздуха примем равным 760 мм ртутного столба, отсюда найдем плотность воздуха...*

Если тот, кто поставил задачу на "человеческом" языке не будет дальше вмешиваться в ход мысли математика, то последний через некоторое время даст численный ответ. Но "потребитель" может возражать по-прежнему: *камень на самом деле вовсе не эллипсоидальный, давление воздуха в том месте и в тот момент не было равно 760 мм ртутного столба и т.д.* Что же ответит ему математик?

Он ответит: *"Точное решение реальной задачи вообще невозможно. Мало того, что форму камня, которая влияет на сопротивление воздуха, невозможно описать никаким математическим уравнением; его вращение в полете также неподвластно математике из-за своей сложности. Далее, воздух не является однородным, так как в результате действия случайных факторов в нем возникают флуктуации колебания плотности. Если пойти ещё глубже, нужно учесть, что по закону всемирного тяготения каждое тело действует на каждое другое тело. Отсюда следует, что даже маятник настенных часов изменяет своим движением траекторию камня.*

Короче говоря, если мы всерьез захотим точно исследовать поведение какого-либо предмета, то нам предварительно придется узнать местонахождение и скорость всех остальных предметов Вселенной. А это, разумеется, невозможно

Чтобы описать явление, необходимо выявить самые существенные его свойства, закономерности, внутренние связи, роль отдельных характеристик явления. Выделив наиболее важные факторы, можно пренебречь менее существенными.

Наиболее эффективно математическую модель можно реализовать на компьютере в виде алгоритмической модели — так называемого "вычислительного эксперимента" (см. [1], параграф 26).

Конечно, результаты вычислительного эксперимента могут оказаться и не соответствующими действительности, если в модели не будут учтены какие-то важные стороны действительности.

Итак, создавая математическую модель для решения задачи, нужно:

1. выделить предположения, на которых будет основываться математическая модель;
2. определить, что считать исходными данными и результатами;
3. записать математические соотношения, связывающие результаты с исходными данными.

При построении математических моделей далеко не всегда удается найти формулы, явно выражающие искомые величины через данные. В таких случаях используются математические методы, позволяющие дать ответы той или иной степени точности.

Существует не только математическое моделирование какого-либо явления, но и визуально-натурное моделирование, которое обеспечивается за счет отображения этих явлений средствами машинной графики, т.е. перед исследователем демонстрируется своеобразный "компьютерный мультфильм", снимаемый в реальном масштабе времени. Наглядность здесь очень высока.

8.3. Какие основные этапы содержит процесс разработки программ?

Процесс разработки программы можно выразить следующей формулой:

$$\boxed{\text{РАЗРАБОТКА ПРОГРАММЫ}} = \boxed{\text{ИЗГОТОВЛЕНИЕ}} + \boxed{\text{ДОКАЗАТЕЛЬСТВО ПРАВИЛЬНОСТИ}}$$

На начальном этапе работы анализируются и формулируются требования к программе, разрабатывается точное описание того, что должна делать программа и каких результатов необходимо достичь с ее помощью.

Затем программа разрабатывается с использованием той или иной технологии программирования (например, структурного программирования).

Полученный вариант программы подвергается систематическому тестированию — ведь наличие ошибок в только что разработанной программе это вполне нормальное закономерное явление. Практически невозможно составить реальную (достаточно сложную) программу без ошибок. Нельзя делать вывод, что программа правильна, лишь на том основании, что она не отвергнута машиной и выдала результаты. Все, что достигнуто в этом случае, это получение каких-то результатов, не обязательно

правильных. В программе при этом может оставаться большое количество логических ошибок. Ответственные участки программы проверяются с использованием методов доказательства правильности программ.

Для каждой программы обязательно проводятся работы по обеспечению качества и эффективности программного обеспечения, анализируются и улучшаются временные характеристики.

8.4. Как проконтролировать текст программы до выхода на компьютер?

Текст программы можно проконтролировать за столом с помощью просмотра, проверки и прокрутки.

- **Просмотр.** Текст программы просматривается на предмет **обнаружения описок и расхождений с алгоритмом**. Нужно просмотреть **организацию всех циклов**, чтобы убедиться в правильности операторов, задающих кратности циклов. Полезно посмотреть еще раз **условия в условных операторах, аргументы в обращениях к подпрограммам** и т.п.
- **Проверка.** При проверке программы программист по тексту программы мысленно старается восстановить тот вычислительный процесс, который определяет программа, после чего сверяет его с требуемым процессом. На время проверки нужно **"забыть"**, что должна делать программа, и **"узнавать"** об этом по ходу её проверки. Только после окончания проверки программы можно **"вспомнить"** о том, что она должна делать и **сравнить** реальные действия программы с требуемыми.
- **Прокрутка.** Основой прокрутки является **имитация программистом за столом выполнения программы на машине**. Для выполнения прокрутки приходится задаваться какими-то исходными данными и производить над ними необходимые вычисления. **Прокрутка — трудоемкий процесс**, поэтому ее следует применять лишь для контроля логически сложных участков программ. **Исходные данные должны выбираться такими, чтобы в прокрутку вовлекалось большинство ветвей программы.**

8.5. Для чего нужны отладка и тестирование?

Отладка программы — это процесс поиска и устранения ошибок в программе, производимый по результатам её прогона на компьютере.

Тестирование (англ. test — испытание) — это испытание, проверка правильности работы программы в целом, либо её составных частей.

Отладка и тестирование — это два четко различимых и непохожих друг на друга этапа:

- при **отладке** происходит локализация и устранение синтаксических ошибок и явных ошибок кодирования;

- в процессе же **тестирования** проверяется работоспособность программы, не содержащей явных ошибок.

Тестирование устанавливает факт наличия ошибок, а отладка выясняет ее причину.

Английский термин *debugging* ("отладка") буквально означает "вылавливание жучков". Термин появился в 1945 г., когда один из первых компьютеров — "Марк-1" прекратил работу из-за того, что в его электрические цепи попал мотылек и заблокировал своими останками одно из тысяч реле машины.

8.6. В чем заключается отладка?

В современных программных системах (Turbo Basic, Turbo Pascal, Turbo C и др.) отладка осуществляется часто с использованием специальных программных средств, называемых отладчиками. Эти средства позволяют **исследовать внутреннее поведение программы**.

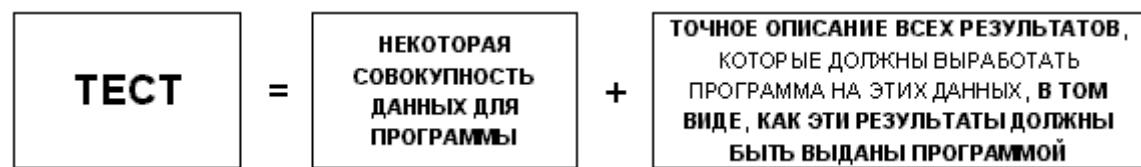
Программа-отладчик обычно обеспечивает следующие возможности:

- **пошаговое исполнение программы** с остановкой после каждой команды (оператора);
- **просмотр текущего значения любой переменной или нахождение значения любого выражения**, в том числе, с использованием стандартных функций; при необходимости можно установить новое значение переменной;
- **установку в программе "контрольных точек"**, т.е. точек, в которых программа временно прекращает свое выполнение, так что можно оценить промежуточные результаты, и др.

При отладке программ **важно помнить следующее:**

- в начале процесса отладки надо использовать **простые тестовые данные**;
- возникающие затруднения следует четко **разделять и устранять строго поочередно**;
- **не нужно считать причиной ошибок машину**, так как современные машины и трансляторы обладают чрезвычайно высокой надежностью.

8.7. Что такое тест и тестирование?



Как бы ни была тщательно отлажена программа, решающим этапом, устанавливающим ее пригодность для работы, является контроль программы по результатам ее выполнения на системе тестов.

Программу условно можно считать правильной, если её запуск для выбранной системы тестовых исходных данных во всех случаях дает правильные результаты.

Но, как справедливо указывал известный теоретик программирования Э. Дейкстра, тестирование может показать лишь наличие ошибок, но не их отсутствие. Нередки случаи, когда новые входные данные вызывают "отказ" или получение неверных результатов работы программы, которая считалась полностью отлаженной.

Для реализации метода тестов должны быть изготовлены или заранее известны эталонные результаты.

Вычислять эталонные результаты нужно обязательно до, а не после получения машинных результатов.

В противном случае имеется опасность невольной подгонки вычисляемых значений под желаемые, полученные ранее на машине.

8.8. Какими должны быть тестовые данные?

Тестовые данные должны обеспечить проверку всех возможных условий возникновения ошибок:

- должна быть испытана **каждая ветвь** алгоритма;
- очередной тестовый прогон должен контролировать нечто такое, что еще **не было проверено** на предыдущих прогонах;
- первый тест должен быть **максимально прост**, чтобы проверить, работает ли программа вообще;
- арифметические операции в тестах должны **предельно упрощаться** для уменьшения объема вычислений;
- количества элементов последовательностей, точность для итерационных вычислений, количество проходов цикла в тестовых примерах должны задаваться из соображений **сокращения объема вычислений**;
- минимизация вычислений не должна снижать надежности контроля;
- тестирование должно быть **целенаправленным и систематизированным**, так как случайный выбор исходных данных привел бы к трудностям в определении ручным способом ожидаемых результатов; кроме того, при случайном выборе тестовых данных могут оказаться непроверенными многие ситуации;
- **усложнение** тестовых данных должно происходить **постепенно**.

Пример. Система тестов для задачи нахождения корней квадратного уравнения $ax^2 + bx + c = 0$:

Номер теста	Проверяемый случай	Коэффициенты			Результаты
		a	b	c	

1	$d > 0$	1	1	-2	$x_1 = 1, x_2 = -2$
2	$d = 0$	1	2	1	Корни равны: $x_1 = -1, x_2 = -1$
3	$d < 0$	2	1	2	Действительных корней нет
4	$a=0, b=0, c=0$	0	0	0	Все коэффициенты равны нулю. x — любое число.
5	$a=0, b=0, c \neq 0$	0	0	2	Неправильное уравнение
6	$a=0, b \neq 0$	0	2	1	Линейное уравнение. Один корень: $x = -0,5$
7	$a \neq 0, b \neq 0, c = 0$	2	1	0	$x_1 = 0, x_2 = -0,5$

8.9. Из каких этапов состоит процесс тестирования?

Процесс тестирования можно разделить на три этапа.

1. Проверка в нормальных условиях. Предполагает тестирование на основе данных, которые характерны для реальных условий функционирования программы.

2. Проверка в экстремальных условиях. Тестовые данные включают **граничные значения** области изменения входных переменных, которые должны восприниматься программой как правильные данные. Типичными примерами таких значений являются очень маленькие или очень большие числа и отсутствие данных. Еще один тип экстремальных условий — это **граничные объемы** данных, когда массивы состоят из слишком малого или слишком большого числа элементов.

3. Проверка в исключительных ситуациях. Проводится с использованием данных, значения которых лежат **за пределами допустимой области изменений**. Известно, что все программы разрабатываются в расчете на обработку **какого-то ограниченного набора данных**. Поэтому важно получить ответ на следующие вопросы:

- что произойдет, если программе, не рассчитанной на обработку отрицательных и нулевых значений переменных, в результате какой-либо ошибки придется иметь дело как раз с такими данными?
- как будет вести себя программа, работающая с массивами, если количество их элементов превысит величину, указанную в объявлении массива?
- что произойдет, если числа будут слишком малыми или слишком большими?

Наихудшая ситуация складывается тогда, когда программа воспринимает неверные данные как правильные и выдает неверный, но правдоподобный результат.

Программа должна сама отвергать любые данные, которые она не в состоянии обрабатывать правильно.

8.10. Каковы характерные ошибки программирования?

Ошибки могут быть допущены на всех этапах решения задачи — от ее постановки до оформления. Разновидности ошибок и соответствующие примеры приведены в таблице:

Вид ошибки	Пример
Неправильная постановка задачи	Правильное решение неверно сформулированной задачи
Неверный алгоритм	Выбор алгоритма, приводящего к неточному или эффективному решению задачи
Ошибка анализа	Неполный учет ситуаций, которые могут возникнуть; логические ошибки
Семантические ошибки	Непонимание порядка выполнения оператора
Синтаксические ошибки	Нарушение правил, определяемых языком программирования
Ошибки при выполнении операций	Слишком большое число, деление на ноль, извлечение квадратного корня из отрицательного числа и т. п.
Ошибки в данных	Неудачное определение возможного диапазона изменения данных
Опечатки	Перепутаны близкие по написанию символы, например, цифра 1 и буквы <i>l</i> , <i>l</i>
Ошибки ввода-вывода	Неверное считывание входных данных, неверное задание форматов данных

8.11. Является ли отсутствие синтаксических ошибок свидетельством правильности программы?

Обычно синтаксические ошибки выявляются на этапе трансляции. Многие же другие ошибки транслятору выявить невозможно, так как **транслятору неизвестны замыслы программиста.**

Отсутствие сообщений машины о синтаксических ошибках является необходимым, но не достаточным условием, чтобы считать программу правильной.

Примеры синтаксических ошибок:

- пропуск знака пунктуации;
- несогласованность скобок;
- неправильное формирование оператора;
- неверное образование имен переменных;
- неверное написание служебных слов;

- отсутствие условий окончания цикла;
- отсутствие описания массива и т.п.

8.12. Какие ошибки не обнаруживаются транслятором?

Существует множество ошибок, которые транслятор выявить не в состоянии, если используемые в программе операторы сформированы верно. Приведем примеры таких ошибок.

Логические ошибки:

- неверное указание ветви алгоритма после проверки некоторого условия;
- неполный учет возможных условий;
- пропуск в программе одного или более блоков алгоритма.

Ошибки в циклах:

- неправильное указание начала цикла;
- неправильное указание условий окончания цикла;
- неправильное указание числа повторений цикла;
- бесконечный цикл.

Ошибки ввода-вывода; ошибки при работе с данными:

- неправильное задание тип данных;
- организация считывания меньшего или большего объема данных, чем требуется;
- неправильное редактирование данных.

Ошибки в использовании переменных:

- использование переменных без указания их начальных значений;
- ошибочное указание одной переменной вместо другой.

Ошибки при работе с массивами:

- массивы предварительно не обнулены;
- массивы неправильно описаны;
- индексы следуют в неправильном порядке.

Ошибки в арифметических операциях:

- неверное указание типа переменной (например, целочисленного вместо вещественного);
- неверное определение порядка действий;
- деление на нуль;
- извлечение квадратного корня из отрицательного числа;
- потеря значащих разрядов числа.

Все эти ошибки обнаруживаются с помощью тестирования.

8.13. В чем заключается сопровождение программы?

Сопровождение программ — это работы, связанные с обслуживанием программ в процессе их эксплуатации.

Многokратное использование разработанной программы для решения различных задач заданного класса требует проведения следующих дополнительных работ:

- исправление обнаруженных ошибок;
- модификация программы для удовлетворения изменяющихся эксплуатационных требований;
- доработка программы для решения конкретных задач;
- проведение дополнительных тестовых просчетов;
- внесение исправлений в рабочую документацию;
- усовершенствование программы и т.д.

Применительно ко многим программам работы по сопровождению поглощают более половины затрат, приходящихся на весь период времени существования программы (начиная от выработки первоначальной концепции и кончая моральным ее устареванием) в стоимостном выражении.

Программа, предназначенная для длительной эксплуатации, должна иметь соответствующую документацию и инструкцию по её использованию.

8.14. Вопросы для самоконтроля

- 8.1. Какие основные этапы включает в себя решение задач на компьютере?
- 8.2. Какие этапы компьютерного решения задач осуществляются без участия компьютера?
- 8.3. Что называют математической моделью объекта или явления?
- 8.4. Почему невозможно точное исследование поведения объектов или явлений?
- 8.5. Какие способы моделирования осуществляются с помощью компьютера?
- 8.6. Из каких последовательных действий состоит процесс разработки программы?
- 8.7. Доказывает ли получение правдоподобного результата правильность программы?
- 8.8. Какие ошибки могут остаться невыявленными, если не провести проверку (просмотр, прокрутку) программы?
- 8.9. Чем тестирование программы отличается от её отладки?
- 8.10. Каким образом программа-отладчик помогает исследовать поведение программы в процессе её выполнения?

- 8.11. Как следует планировать процесс отладки программы?
- 8.12. Можно ли с помощью тестирования доказать правильность программы?
- 8.13. На какой стадии работы над программой вычисляются эталонные результаты тестов?
- 8.14. Назовите основные этапы процесса тестирования.
- 8.15. В чём заключается отличие синтаксических ошибок от семантических?
- 8.16. О чём свидетельствует отсутствие сообщений машины о синтаксических ошибках?
- 8.17. Какие разновидности ошибок транслятор не в состоянии обнаружить?
- 8.18. Для чего программам требуется сопровождение?

8.15. Упражнения

Составьте системы тестов для решения следующих задач:

- 8.1. Найдите наибольший общий делитель двух заданных целых чисел.
- 8.2. Найдите наименьшее общее кратное двух заданных целых чисел.
- 8.3. Определите, является ли заданное число нечетным двузначным числом.
- 8.4. Заданы площади квадрата и круга. Определите, поместится ли квадрат в круге.
- 8.5. Решите биквадратное уравнение.
- 8.6. Найдите среднее арифметическое положительных элементов заданного одномерного массива.
- 8.7. Элементы заданного одномерного массива разделите на его первый элемент.
- 8.8. Определите, лежит ли заданная точка на одной из сторон треугольника, заданного координатами своих вершин.
- 8.9. Определите, имеют ли общие точки две плоские фигуры — треугольник с заданными координатами его вершин и круг заданного радиуса с центром в начале координат.
- 8.10. Задано целое $A > 1$. Найдите наименьшее целое неотрицательное k , при котором $2^k > A$.
- 8.11. Дана последовательность целых чисел. Определите, со скольких чётных чисел она начинается.
- 8.12. В заданном двумерном массиве найдите количество строк, не содержащих нули.

8.13. Определите, сколько строк заданного двумерного массива содержат элементы из заданного диапазона.

8.14. Преобразуйте число, заданное в римской системе счисления, в число десятичной системы.

Глава 9. Применения информатики и компьютерной техники

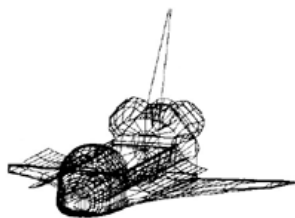
9.1. Как используются компьютеры в быту?

В последнее время компьютеры <проникли> в жилища людей и постепенно становятся предметами первой необходимости. Есть два основных направления использования компьютеров дома.

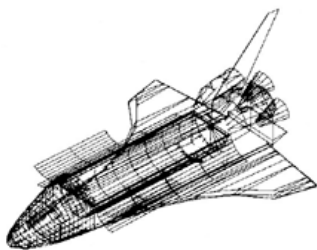
1. **Обеспечение нормальной жизнедеятельности жилища:**
 - охранная автоматика, противопожарная автоматика, газоанализаторная автоматика;
 - управление освещенностью, расходом электроэнергии, отопительной системой, управление микроклиматом;
 - электроплиты, холодильники, стиральные машины со встроенными микропроцессорами.
2. **Обеспечение информационных потребностей людей, находящихся в жилище:**
 - заказы на товары и услуги;
 - [процессы обучения](#);
 - общение с [базами данных](#) и [знаний](#);
 - сбор данных о [состоянии здоровья](#);
 - обеспечение досуга и развлечений;
 - обеспечение справочной информацией;
 - [электронная почта](#), [телеконференции](#);
 - [Интернет](#).

9.2. Что такое системы автоматизированного проектирования (САПР)?

Системы автоматизированного проектирования (САПР) — комплексные программно-технические системы, предназначенные для выполнения проектных работ с применением математических методов.



Системы САПР широко используются в архитектуре, электронике, энергетике, механике и др. В процессе автоматизированного проектирования в качестве входной информации используются технические знания специалистов, которые вводят проектные требования, уточняют результаты, проверяют полученную конструкцию, изменяют ее и т.д.



Кроме того, в САПР накапливается информация, поступающая из библиотек стандартов (данные о типовых элементах конструкций, их размерах, стоимости и др.). В процессе проектирования разработчик вызывает определенные программы и выполняет их. Из САПР информация выдается в виде готовых комплектов законченной технической и проектной документации.

9.3. Что такое автоматизированные системы научных исследований (АСНИ)?

Автоматизированные системы научных исследований (АСНИ) предназначены для автоматизации научных экспериментов, а также для осуществления моделирования исследуемых объектов, явлений и процессов, изучение которых традиционными средствами затруднено или невозможно.

В настоящее время научные исследования во многих областях знаний проводят большие коллективы ученых, инженеров и конструкторов с помощью весьма сложного и дорогого оборудования.

Большие затраты ресурсов для проведения исследований обусловили необходимость повышения эффективности всей работы.

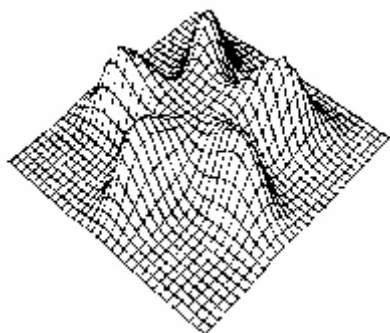
Эффективность научных исследований в значительной степени связана с уровнем использования компьютерной техники.

Компьютеры в АСНИ используются в информационно-поисковых и экспертных системах, а также решают следующие задачи:

- управление экспериментом;
- подготовка отчетов и документации;
- поддержание базы экспериментальных данных и др.

В результате применения АСНИ возникают следующие положительные моменты:

- в несколько раз сокращается время проведения исследования;
- увеличивается точность и достоверность результатов;
- усиливается контроль за ходом эксперимента;
- сокращается количество участников эксперимента;
- повышается качество и информативность эксперимента за счет увеличения числа контролируемых параметров и более тщательной обработки данных;



- результаты экспериментов выводятся оперативно в наиболее удобной форме — графической или символьной (например, значения функции многих переменных выводятся средствами машинной графики в виде так называемых «горных массивов»). На экране одного графического монитора возможно формирование целой системы приборных шкал (вольтметров, амперметров и др.), регистрирующих параметры экспериментального объекта.

9.4. Какая взаимосвязь между АСНИ и САПР?

Каждая из систем АСНИ и САПР, конечно, имеет свою специфику и отличается поставленными целями и методами их достижения. Однако очень часто между обоими

типами систем обнаруживается тесная связь, и их роднит не только то, что они реализуются на базе компьютерной техники.

Например, в процессе проектирования может потребоваться выполнение того или иного исследования, и, наоборот, в ходе научного исследования может возникнуть потребность и в конструировании нового прибора и в проектировании научного эксперимента.

Такая взаимосвязь приводит к тому, что на самом деле «чистых» АСНИ и САПР не бывает: в каждой из них можно найти общие элементы. С повышением их интеллектуальности они сближаются. В конечном счете и те и другие должны представлять собой экспертную систему, ориентированную на решение задач конкретной области.

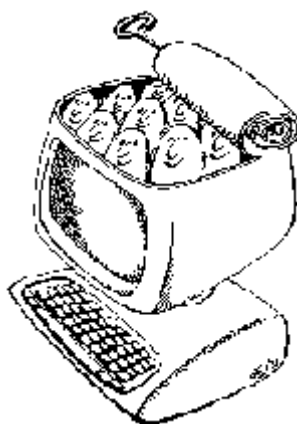
9.5. Что такое базы знаний и экспертные системы?

База знаний (*knowledge base*) — совокупность знаний, относящихся к некоторой предметной области и формально представленных таким образом, чтобы на их основе можно было осуществлять рассуждения.

Базы знаний чаще всего используются в контексте *экспертных систем*, где с их помощью представляются *навыки и опыт экспертов*, занятых практической деятельностью в соответствующей области (например, в медицине или в математике). Обычно база знаний представляет собой *совокупность правил вывода*.

Экспертная система — это комплекс компьютерного программного обеспечения, помогающий человеку принимать обоснованные решения. Экспертные системы используют информацию, полученную заранее от *экспертов* — людей, которые в какой-либо области являются лучшими специалистами.

Экспертные системы должны:



- хранить знания об определенной предметной области (факты, описания событий и закономерностей);
- уметь общаться с пользователем на ограниченном естественном языке (т.е. задавать вопросы и понимать ответы);

- обладать комплексом логических средств для выведения новых знаний, выявления закономерностей, обнаружения противоречий;
- ставить задачу по запросу, уточнять её постановку и находить решение;
- объяснять пользователю, каким образом получено решение.

Желательно также, чтобы экспертная система могла:

- сообщать такую информацию, которая повышает доверие пользователя к экспертной системе;
- «рассказывать» о себе, о своей собственной структуре.

Экспертные системы могут использоваться в различных областях — *медицинской диагностике*, при *поиске неисправностей*, *разведке полезных ископаемых*, *выборе архитектуры компьютерной системы* и т.д.

9.6. Как используются компьютеры в административном управлении?

Основные применения компьютеров в административном управлении следующие.

Электронный офис. Это система автоматизации работы учреждения, основанная на использовании компьютерной техники. В нее обычно входят такие компоненты, как:

- [текстовые редакторы](#);
- [интегрированные пакеты](#) программ;
- [электронные таблицы](#);
- [системы управления базами данных](#);
- [графические редакторы](#) и графические библиотеки (для получения диаграмм, схем, графиков и др.);
- [электронные записные книжки](#);
- [электронные календари](#) с расписанием деловых встреч, заседаний и др.;
- электронные картотеки, обеспечивающие каталогизацию и поиск документов (писем, отчетов и др.) с помощью компьютера;
- автоматические телефонные справочники, которые можно листать на экране, установить курсором нужный номер и соединиться.

Более подробно многие из перечисленных компонент описаны в главе 6 «Программное обеспечение».

Автоматизация документооборота с использованием специальных электронных устройств:

- ***адаптера*** (лат. *adaptare* — приспособлять) ***связи с периферийными устройствами***, имеющего выход на телефонную линию;
- ***сканера*** (англ. *scan* — поле зрения) для ввода в компьютер документов — текстов, чертежей, графиков, рисунков, фотографий.

Электронная почта. Это система пересылки сообщений между пользователями вычислительных систем, в которой компьютер берет на себя все функции по хранению и пересылке сообщений. Для осуществления такой пересылки отправитель и получатель не

обязательно должны одновременно находиться у дисплеев и не обязательно должны быть подключены к одному компьютеру.

Отправитель сообщения прежде всего запускает программу отправки почты и создает файл сообщения. Затем это сообщение передается в систему пересылки сообщений, которая отвечает за его доставку адресатам. Спустя некоторое время сообщение доставляется адресату и помещается в его «*почтовый ящик*», размещенный на магнитном диске. Затем получатель запускает программу, которая извлекает полученные сообщения, заносит их в архив и т.п.

Система контроля исполнения приказов и распоряжений.

Система телеконференций. Это основанная на использовании компьютерной техники система, позволяющая пользователям, несмотря на их взаимную удаленность в пространстве, а иногда, и во времени, участвовать в совместных мероприятиях, таких, как организация и управление сложными проектами.

Пользователи обеспечиваются терминалами (обычно это дисплеи и клавиатуры), подсоединенными к компьютеру, которые позволяют им связываться с другими членами группы. Для передачи информации между участниками совещания используются *линии связи*.

Работа системы регулируется *координатором*, в функции которого входит организация работы участников совещания, обеспечение их присутствия на совещании и передача сообщаемой ими информации другим участникам совещания.

В некоторых системах телеконференцсвязи участники имеют возможность «видеть» друг друга, что обеспечивается подсоединенными к системам телевизионными камерами и дисплеями.

9.7. Какую роль играют компьютеры в обучении?

Процесс подготовки квалифицированных специалистов длителен и сложен. Обучение в средней школе и затем в вузе занимает почти треть продолжительности жизни человека. К тому же в современном информационном обществе знания очень быстро стареют. Чтобы быть способным выполнять ту или иную профессиональную деятельность, специалисту необходимо непрерывно пополнять своё образование.

**В информационном обществе знать "КАК" важнее, чем
знать "ЧТО".**

Поэтому в наше время **основная задача среднего и высшего этапов образования состоит не в том, чтобы сообщить как можно больший объем знаний, а в том, чтобы научить эти знания добывать самостоятельно и творчески применять для получения нового знания.** Реально это возможно лишь с введением в образовательный процесс средств новых информационных технологий (СНИТ), ориентированных на реализацию целей обучения и воспитания.

Средства новых информационных технологий — это программно-аппаратные средства и устройства, функционирующие на базе ком-пьютерной техники, а также современные средства и системы информационного обмена, обеспечивающие операции по сбору, созданию, накоплению, хранению, обработке и передачи информации.

Рассмотрим **основные перспективные направления использования СНИТ в образовании** [52, 53].

1. Автоматизированные обучающие системы (АОС) — комплексы программно-технических и учебно-методических средств, обеспечивающих активную учебную деятельность. АОС обеспечивают не только обучение конкретным знаниям, но и проверку ответов учащихся, возможность подсказки, занимательность изучаемого материала и др.

АОС представляют собой сложные человеко-машинные системы, в которых объединяется в одно целое ряд дисциплин: **дидактика** (научно обосновываются цели, содержание, закономерности и принципы обучения); **психология** (учитываются особенности характера и душевный склад обучаемого); **моделирование, машинная графика** и др.

Основное средство взаимодействия обучаемого с АОС — **диалог**. Диалогом с обучающей системой может управлять как сам обучаемый, так и система. В первом случае обучаемый сам определяет режим своей работы с АОС, выбирая способ изучения материала, который соответствует его индивидуальным способностям. Во втором случае методику и способ изучения материала выбирает система, предъявляя обучаемому в соответствии со сценарием кадры учебного материала и вопросы к ним. Свои ответы обучаемый вводит в систему, которая истолковывает для себя их смысл и выдает сообщение о характере ответа. В зависимости от степени правильности ответа, либо от вопросов обучаемого система организует запуск тех или иных путей сценария обучения, выбирая стратегию обучения и приспособляясь к уровню знаний обучаемого.

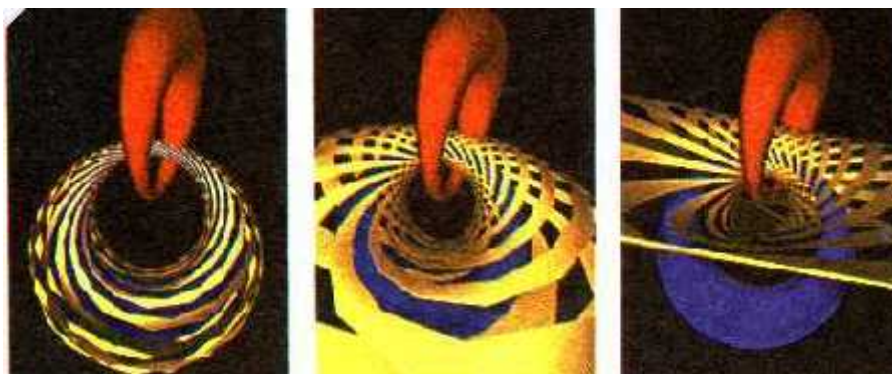
2. Экспертные обучающие системы (ЭОС). Реализуют обучающие функции и содержат знания из определенной достаточно узкой предметной области. ЭОС располагают возможностями пояснения стратегии и тактики решения задачи изучаемой предметной области и обеспечивают контроль уровня знаний, умений и навыков с диагностикой ошибок по результатам обучения.

3. Учебные базы данных (УБД) и учебные базы знаний (УБЗ), ориентированные на некоторую предметную область. УБД позволяют формировать наборы данных для заданной учебной задачи и осуществлять выбор, сортировку, анализ и обработку содержащейся в этих наборах информации. В УБЗ, как правило, содержатся описание основных понятий предметной области, стратегия и тактика решения задач; комплекс предлагаемых упражнений, примеров и задач предметной области, а также перечень возможных ошибок обучаемого и информация для их исправления; база данных, содержащая перечень методических приемов и организационных форм обучения.

4. Системы Мультимедиа. Позволяют реализовать интенсивные методы и формы обучения, повысить мотивацию обучения за счет применения современных средств обработки аудиовизуальной информации, повысить уровень эмоционального

восприятия информации, сформировать умения реализовывать разнообразные формы самостоятельной деятельности по обработке информации.

Системы Мультимедиа широко используются с целью изучения процессов различной природы на основе их моделирования. Здесь можно сделать наглядной невидимую обычным глазом жизнь элементарных частиц микромира при изучении физики, образно и понятно рассказать об абстрактных и n-мерных мирах, доходчиво объяснить, как работает тот или иной алгоритм и т.п. Возможность **в цвете и со звуковым сопровождением промоделировать реальный процесс** поднимает обучение на качественно новую ступень.



Кадры учебного компьютерного фильма "Топология и механика" изображают гиперсферу, аналог обычной сферы в четырехмерном пространстве.

5. Системы <Виртуальная реальность>. Применяются при решении конструктивно-графических, художественных и других задач, где необходимо развитие умения создавать мысленную пространственную конструкцию некоторого объекта по его графическому представлению; при изучении стереометрии и черчения; в компьютеризированных тренажерах технологических процессов, ядерных установок, авиационного, морского и сухопутного транспорта, где без подобных устройств принципиально невозможно отработать навыки взаимодействия человека с современными сверхсложными и опасными механизмами и явлениями.

6. Образовательные компьютерные телекоммуникационные сети. Позволяют обеспечить дистанционное обучение (ДО) — обучение на расстоянии, когда преподаватель и обучаемый разделены пространственно и (или) во времени, а учебный процесс осуществляется с помощью телекоммуникаций, главным образом, на основе средств сети Интернет. Многие люди при этом получают возможность повышать образование на дому (например, взрослые люди, обремененные деловыми и семейными заботами, молодежь, проживающая в сельской местности или небольших городах). Человек в любой период своей жизни обретает возможность дистанционно получить новую профессию, повысить свою квалификацию и расширить кругозор, причем практически в любом научном или учебном центре мира.

В образовательной практике находят применение все основные виды компьютерных телекоммуникаций: электронная почта, электронные доски объявлений, телеконференции и другие возможности Интернета. ДО предусматривает и автономное использование курсов, записанных на видеодиски, компакт-диски и т.д. Компьютерные телекоммуникации обеспечивают:

- возможность доступа к различным источникам информации через систему Internet и работы с этой информацией;

- возможность оперативной обратной связи в ходе диалога с преподавателем или с другими участниками обучающего курса;
- возможность организации совместных телекоммуникационных проектов, в том числе международных, телеконференций, возможность обмена мнениями с любым участником данного курса, преподавателем, консультантами, возможность запроса информации по любому интересующему вопросу через телеконференции.
- возможность реализации методов дистанционного творчества, таких как участие в дистанционных конференциях, дистанционный <мозговой штурм> сетевых творческих работ, сопоставительный анализ информации в WWW, дистантные исследовательские работы, коллективные образовательные проекты, деловые игры, практикумы, виртуальные экскурсии др.

Совместная работа стимулирует учащихся на ознакомление с разными точками зрения на изучаемую проблему, на поиск дополнительной информации, на оценку получаемых собственных результатов.

9.8. Какую роль играют компьютеры в управлении технологическими процессами?

Основных применений два:

- в гибких автоматизированных производствах (ГАП);
- в контрольно-измерительных комплексах.

В гибких автоматизированных производствах компьютеры (или микропроцессоры) решают следующие задачи:

- управление механизмами;
- управление технологическими режимами;
- управление промышленными роботами.

Применение компьютеров в управлении технологическими процессами оправдано тогда, когда существует потребность в частых изменениях реализуемых функций. Пример гибких автоматизированных производств — **заводы-роботы** в Японии.

Одной из новых областей является создание на основе персональных компьютеров **контрольно-измерительной аппаратуры**, с помощью которой можно проверять изделия прямо **на производственной линии**.

В развитых странах налажен выпуск программного обеспечения и специальных сменных плат, позволяющих **превращать компьютер в высококачественную измерительную и испытательную систему**.

Компьютеры, оснащенные подобным образом, могут использоваться в качестве **запоминающих цифровых осциллографов, устройств сбора данных, многоцелевых измерительных приборов**.

Применение компьютеров в качестве контрольно-измерительных приборов более эффективно, чем выпуск в ограниченных количествах специализированных приборов с вычислительными блоками.

Автоматизированное рабочее место (АРМ, рабочая станция) — место оператора, которое оборудовано всеми средствами, необходимыми для выполнения определенных функций.

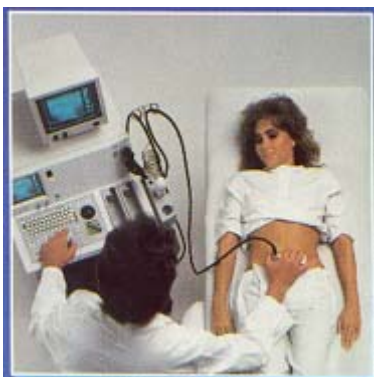
В системах обработки данных и учреждениях обычно АРМ — это дисплей с клавиатурой, но может использоваться также и принтер, внешние запоминающее устройство и др.

9.9. Какую роль играют компьютеры в медицине?

Врачи используют компьютеры для многих важных применений. Назовем некоторые из них.

Компьютерная аппаратура широко используется при постановке диагноза, проведении обследований и профилактических осмотров. Примеры компьютерных устройств и методов лечения и диагностики:

- **компьютерная томография и ядерная медицинская диагностика** — дают точные послойные изображения структур внутренних органов;



- **ультразвуковая диагностика и зондирование** — используя эффекты взаимодействия падающих и отраженных ультразвуковых волн, открывает бесчисленные возможности для получения изображений внутренних органов и исследования их состояния;

- **микрокомпьютерные технологии рентгеновских исследований** — запомненные в цифровой форме рентгеновские снимки могут быть быстро и качественно обработаны, воспроизведены и занесены в архив для сравнения с последующими снимками этого пациента;



- **задатчик (водитель) сердечного ритма;**



- **устройства дыхания и наркоза;**
- **лучевая терапия с микропроцессорным управлением** — обеспечивает возможность применения более надежных и щадящих методов облучения;
- **устройства диагностики и локализации почечных и желчных камней**, а также контроля процесса их разрушения при помощи наружных ударных волн (литотрипсия);
- **лечение зубов и протезирование с помощью компьютера;**
- **системы с микрокомпьютерным управлением для интенсивного медицинского контроля пациента.**

Компьютерные сети используются для **пересылки сообщений о донорских органах, в которых нуждаются больные**, ожидающие операции трансплантации.

Банки медицинских данных позволяют медикам быть в курсе последних научных и практических достижений.

Компьютеры позволяют установить, как влияет **загрязненность воздуха** на заболеваемость населения данного района. Кроме того, с их помощью можно изучать **влияние ударов** на различные части тела, в частности, последствия удара при автомобильной катастрофе для черепа и позвоночника человека.

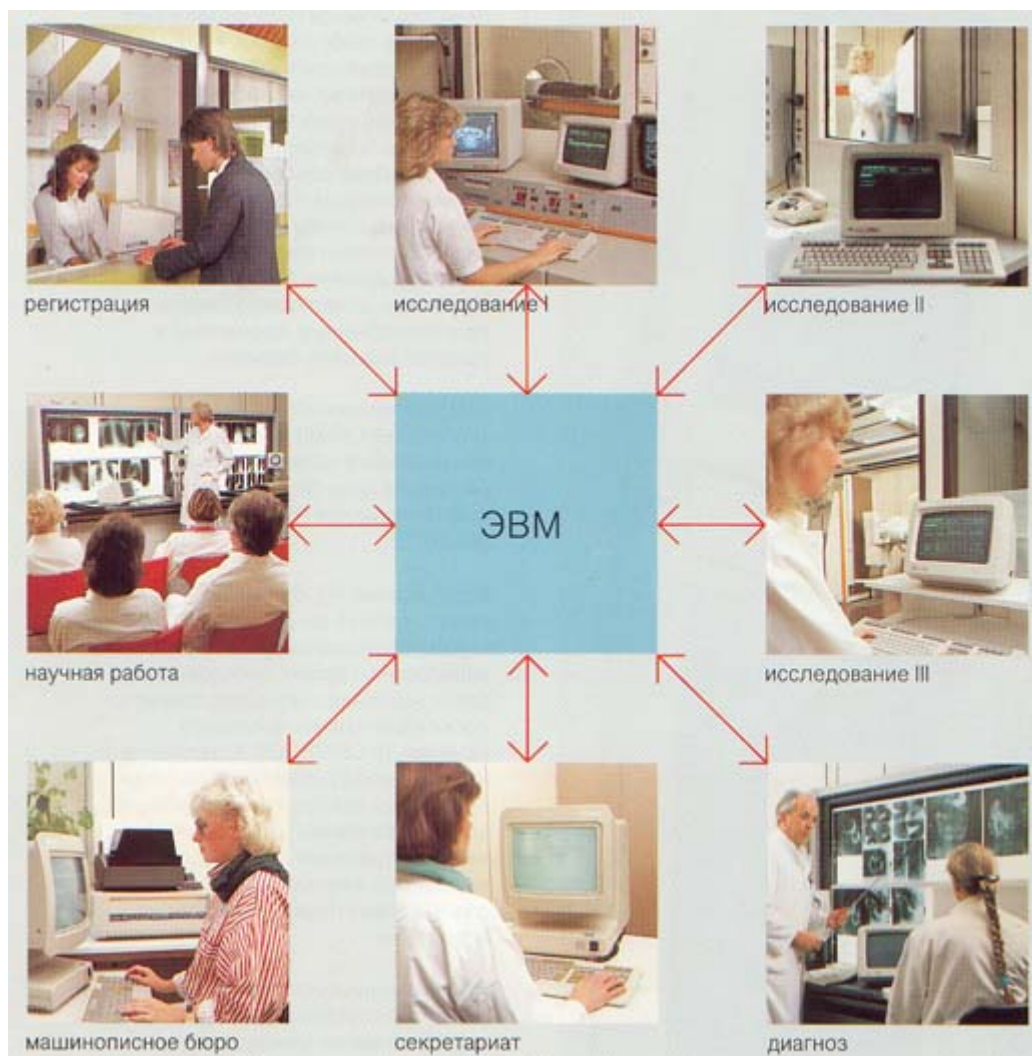


Рис 9.1. Организация работы в компьютеризированном отделении лечебного учреждения

Компьютерная техника используется для **обучения медицинских работников практическим навыкам**. На этот раз компьютер выступает в роли больного, которому требуется немедленная помощь. На основании симптомов, выданных компьютером, обучающийся должен определить курс лечения. Если он ошибся, компьютер сразу показывает это.

Компьютеры используются для **создания карт, показывающих скорость распространения эпидемий**.

Компьютеры **хранят** в своей памяти **истории болезней пациентов**, что освобождает врачей от бумажной работы, на которую уходит много времени, и позволяет больше времени уделять самим больным.

На рисунках показано, как организуется работа в полностью компьютеризированном отделении лечебного учреждения. Применение компьютеров переводит медицину на иной, более высокий качественный уровень и способствует дальнейшему повышению уровня и качества жизни.

9.10. Как используются компьютеры в торговле?

В организации компьютерного обслуживания торговых предприятий большое распространение получил так называемый **штриховой код** (бар-код). Он представляет собой **серию широких и узких линий, в которых зашифрован номер торгового изделия.**



Этот номер записан на этикетке изделия дважды: в форме двух пятизначных чисел и в виде широких и узких линий. **Первые пять цифр указывают фирму-поставщика, а другие пять — номер изделия в номенклатуре выпускаемых данной фирмой товаров.**

Для печати штриховых кодов используются специальные приставки на обычных принтерах. Полученные коды считываются с помощью сканеров, преобразуются в электрические импульсы, переводятся в двоичный код и передаются в память компьютера.

Используя штриховой код, компьютер печатает на выдаваемом покупателю чеке название товара и его цену.

Информация о каждом имеющемся в магазине или на складе товаре занесена в [базу данных](#). По запросу компьютер анализирует:

- количество оставшегося товара;
- правила его налогообложения;
- юридические ограничения на его продажу и др.

Одновременно с подачей сведений о проданном товаре на дисплей кассового аппарата компьютер производит соответствующую **коррекцию (уточнение) товарной ведомости.**

Обычно программное обеспечение устроено так, что сводная информация о наличии товаров выдается управляющему магазина к концу рабочего дня. Вместе с тем **управляющий имеет возможность обновлять данные** об изменении цен и поступлении новых партий товаров.

В перспективе торговля, по-видимому, превратится в **компьютеризованную продажу товаров по заказам.**

То же самое программное обеспечение, которое применяется для организации учета в торговле, можно использовать и для других целей, например для контроля наличия комплектующих изделий на заводской сборочной линии, учета сплавляемых по реке бревен и др.

9.11. Что такое электронные деньги?

Одной из важнейших составляющих информатизации становится переход денежно-кредитной и финансовой сферы к электронным деньгам.

Основные направления использования электронных денег следующие:

Торговля без наличных. Оплата производится с использованием **кредитных карточек**. Имея вместо наличных денег кредитную карточку, покупатель при любой покупке расплачивается не наличными, а **автоматически снимает со своего счета в банке нужную сумму денег и пересылает ее на счет магазина**. Система торговли без наличных **POS** (англ. Points of Sale System — система кассовых автоматов) выполняет следующие функции:

- верификацию кредитных карточек (т.е. удостоверение их подлинности);
- снятие денег со счета покупателя;
- перечисление их на счет продавца.

POS — наиболее массовая и показательная ветвь системы электронных денег. Она способна также обнаруживать малейшие хищения наличных денег и товаров. Сведения на кредитную карточку наносятся **методом магнитной записи**. В каждую кредитную карточку вставлена магнитная карта — носитель информации. На магнитную карту заранее записываются следующие данные:

- номер личного счета;
- название банка;
- страна;
- категория платёжеспособности клиента;
- размер предоставленного кредита и т.д.

Считается, что сами по себе кредитные карточки экономят 10% расходов на оплату товаров.

Разменные автоматы. Они устанавливаются банками только для своих клиентов, которым предварительно выданы кредитные карточки. Клиент вставляет в автомат кредитную карточку и набирает личный код и сумму, которую он желает иметь наличными. Автомат по банковской сети проверяет правильность кода, снимает указанную сумму со счета клиента и выдает её наличными. Часто несколько банков объединяются и создают общую сеть разменных автоматов.

Банковские сделки на дому. За небольшую месячную плату при наличии дома персонального компьютера и модема вкладчик может связываться через телефонную сеть с компьютеризованными банковскими организациями и получать от них богатый набор услуг.

Встречные зачёты. По всему миру активно внедряются электронные системы потребительского кредита и взаимных расчетов между банками по общему итогу. Такие системы реализуются в виде автоматических **клиринговых** (англ. clearing — очистка) вычислительных сетей **АСН** (Automated Clearing House). По сети идут не только банковские документы, но и информация, важная для принятия ответственных финансовых решений.

Оплата устно. Она заключается в оплате счетов по телефону с гарантированным опознаванием кредитора по паролю и голосу. Другой способ состоит в использовании устройства, способного передавать по телефону факсимильные изображения денежных чеков и счетов (факсимильный — от лат. *fac simile*, — точное воспроизведение подписи, документа и т.д. графическими средствами).

Наряду с очевидными и очень большими преимуществами с электронными деньгами связаны и сложные проблемы — от финансовых до правовых ("электронные ограбления", перелив электронных денег из одной страны в другую и др.). Электронные деньги есть неизбежный результат научно-технического прогресса в денежно-кредитной сфере.

9.12. Как применяются компьютеры в сельском хозяйстве?

Имея компьютер, фермер может легко и быстро рассчитать требуемое для посева количество семян и количество удобрений, спланировать свой бюджет и вести учет домашнего скота. Компьютерные системы могут планировать севооборот, рассчитывать график полива сельхозкультур, управлять подачей корма скоту и выполнять много других полезных функций.

На наших глазах происходит технологическая революция в сельском хозяйстве — компьютеры и индивидуальные микродатчики позволяют контролировать состояние и режим каждого отдельного животного и растения. Это высвобождает значительные материальные и людские ресурсы, резко улучшает качество жизни человека.

В качестве примера приведем портативный компьютер **AgGPS 170** компании Trimble, предназначенный для применения в самых тяжелых условиях, сопутствующих сельскохозяйственным работам. Этот компьютер можно использовать как в ручном варианте, так и монтировать в автомобиль. С его помощью можно управлять сельскохозяйственными работами, просматривать карты полей, регистрировать различные данные о состоянии почвы и посаженных сельскохозяйственных культур и др. Основные характеристики компьютера AgGPS 170:

- полностью герметичный и ударостойкий (выдерживает падение с высоты в 1,2 м);
- функционирует в температурном диапазоне от -30 до +60 градусов Цельсия;



Сельскохозяйственный компьютер AgGPS 170

- ресурс работы от батарей — до 40 часов;
- данные сохраняются на съемном картридже памяти;
- работает под управлением ОС MS Windows CE; используется специальное программное обеспечение для сельского хозяйства;

- с помощью компьютера можно записывать рельеф местности и создавать полевые топографические карты, используя данные геоинформационных систем, вычислять площади полей и обрабатывать статистические данные по полевым работам;
- при соединении с системой разбрызгивателей компьютер может регистрировать данные о применении химикатов и автоматически генерировать соответствующие карты и отчеты;
- с помощью соответствующего ПО возможно регистрировать данные со считывателей штрих-кодов, датчиков слежения за состоянием окружающей среды и погоды и др.

9.13. Вопросы для самоконтроля

9.1. Расскажите об основных направлениях использования компьютеров в жилищах людей.

9.2. Как вы представляете себе информационное общество?

9.3. Какие функции обеспечения жизнедеятельности жилища возлагаются сегодня на компьютер?

9.4. Какие виды повседневных информационных потребностей людей обеспечивают компьютеры?

9.5. В чем суть процесса информатизации образования?

9.6. Какие задачи решаются с помощью автоматизированных обучающих систем?

9.7. Что такое дистанционное обучение?

9.8. Назовите основные перспективные направления использования СНИТ в образовании.

9.9. Расскажите об областях применения систем виртуальной реальности.

9.10. Назовите основные компоненты компьютерных офисных технологий.

9.11. Какие преимущества имеет электронная почта по сравнению с обычной почтой?

9.12. Назовите функции координатора системы телеконференций.

9.13. Назовите темы телеконференций, в которых вы хотели бы участвовать.

9.14. Каким вам представляется информационное наполнение баз данных вашего учебного заведения?

9.15. Какие медицинские технологии и способы лечения стали возможны с развитием компьютерной техники?

9.16. Перечислите основные компоненты автоматизированного офиса.

9.17. Что такое POS-система? Какие виды услуг предоставляет система электронных денег?

9.18. Какие функции выполняет система торговли без наличных?

9.19. Как устроена кредитная карточка?

9.20. Какие функции реализуют автоматические клиринговые компьютерные сети?

9.21. Назовите преимущества и недостатки системы компьютеризированных банковских расчетов.

9.22. Какие примеры эффективного применения компьютеров в сельском хозяйстве вы можете назвать?

Словарь основных понятий и терминов

Автоматизированное рабочее место (АРМ, рабочая станция).

Место оператора, оборудованное всеми средствами, необходимыми для выполнения определённых функций. В системах обработки данных и учреждениях обычно АРМ — это дисплей с клавиатурой, но может использоваться также и принтер, внешние ЗУ и др.

Автоматизированные обучающие системы (АОС).

Комплексы программно-технических и учебно-методических средств, обеспечивающих активную учебную деятельность: обучение конкретным знаниям, проверку ответов учащихся, возможность подсказки, занимательность изучаемого материала.

Автоматизированные системы научных исследований (АСНИ).

Предназначены для автоматизации научных экспериментов, а также для осуществления моделирования исследуемых объектов, явлений и процессов, изучение которых традиционными средствами затруднено или невозможно.

Адаптер.

Устройство связи компьютера с периферийными устройствами.

Адрес.

Номер конкретного байта оперативной памяти компьютера.

Алгебра логики.

Раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

Алгоритм.

Заранее заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения решения задачи за конечное число шагов.

Алфавит.

Фиксированный для данного языка набор основных символов, т.е. "букв алфавита", из которых должен состоять любой текст на этом языке. Никакие другие символы в тексте не допускаются.

Антивирусные программы.

Программы, предотвращающие заражение компьютерным вирусом и ликвидирующие последствия заражения.

Арифметико-логическое устройство (АЛУ).

Часть процессора, которая производит выполнение операций, предусмотренных данным компьютером.

Архитектура фон Неймана.

Архитектура компьютера, имеющего одно арифметико-логическое устройство, через которое проходит поток данных, и одно устройство управления, через которое проходит поток команд. См. также ["Принципы фон-Неймана"](#).

Архитектура компьютера.

Логическая организация, структура и ресурсы компьютера, которые может использовать программист. Определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера.

Ассемблер.

См. [Язык ассемблера](#).

ASCII.

Читается "аски". Американский стандартный код обмена информацией. Широко используется для кодирования в виде байта букв, цифр, знаков операций и других компьютерных символов.

Аудиоадаптер (Sound Blaster, звуковая плата).

Специальная электронная плата, которая позволяет записывать звук, воспроизводить его и создавать программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

База данных.

Один или несколько файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации.

Байт.

Группа из восьми битов, рассматриваемая при хранении данных как единое целое.

Библиотека стандартных подпрограмм.

Совокупность подпрограмм, составленных на одном из языков программирования и удовлетворяющих единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм.

Бит.

Наименьшая единица информации в цифровом компьютере, принимающая значения "0" или "1".

Ввод.

Считывание информации с внешнего устройства в память компьютера.

Вентиль.

См. [Логический элемент](#).

Вещественное число.

В информатике — тип данных, содержащий числа, записанные с десятичной точкой и (или) с десятичным порядком.

Видеоадаптер.

Электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея. Содержит видеопамять, регистры ввода-вывода и модуль BIOS. Посылает в дисплей сигналы управления яркостью лучей и сигналы развёртки изображения.

Винчестер.

См. [Накопитель на жёстких магнитных дисках](#).

Вирус компьютерный.

Специально написанная небольшая программа, которая может "приписывать" себя к другим программам для выполнения каких-либо вредных действий — портит файлы, "засоряет" оперативную память.

Внешняя память.

Совокупность запоминающих устройств для длительного хранения данных. В состав внешней памяти входят накопители на гибких и жестких магнитных дисках, оптические и магнито-оптические накопители, накопители на магнитной ленте. Во внешней памяти обычно хранятся архивы программ и данных. Информация, размещенная на внешних носителях, не зависит от того, включен или выключен компьютер.

Второе поколение компьютерной техники.

Машины, созданные в 1955—65 гг. Элементная база — дискретные транзисторные логические элементы. Оперативная память на магнитных сердечниках.

Высокопроизводительные устройства работы с магнитными лентами, магнитные барабаны и диски. Быстродействие — до сотен тысяч операций в секунду, ёмкость памяти — до нескольких десятков тысяч слов. Языки высокого уровня, широкий набор библиотечных программ, мониторные системы, управляющие режимом трансляции и исполнения программ.

Вывод.

Результаты работы программы, выдаваемые компьютером пользователю, другому компьютеру или во внешнюю память.

Выражение.

В языке программирования — запись правила для вычисления некоторого значения. Строится из констант, переменных и указателей функций, объединенных знаками операций.

Гибкий (флоппи) диск.

Круглая пластиковая пластина, покрытая с обеих сторон магнитным окислом и помещенная в защитную оболочку. Используется как носитель небольших объемов информации.

Глобальная сеть (ГВС).

См. [Сеть компьютерная](#).

Графический редактор.

Программа или комплекс программ, позволяющих создавать и редактировать изображения на экране компьютера: рисовать линии, раскрашивать области экрана, создавать надписи различными шрифтами, обрабатывать изображения, полученные с помощью сканеров. Некоторые редакторы обеспечивают возможность получения изображений трёхмерных объектов, их сечений и разворотов.

Графопостроитель.

Устройство для вывода из компьютера информации в виде графиков и чертежей на неподвижную или вращающуюся на барабане бумагу.

Джойстик.

Стержень-ручка, отклонение которой от вертикального положения приводит к передвижению курсора в соответствующем направлении по экрану дисплея. Часто применяется в компьютерных играх.

Диск.

Круглая металлическая или пластмассовая пластина, покрытая магнитным материалом, на которую информация наносится в виде концентрических дорожек, разделённых на секторы.

Дисковод.

Устройство, управляющее вращением магнитного диска, чтением и записью данных на нём.

Дисплей.

Устройство визуального отображения информации (в виде текста, таблицы, рисунка, чертежа и др.) на экране электронно-лучевого прибора.

Драйверы.

Программы, расширяющие возможности операционной системы по управлению устройствами ввода-вывода, оперативной памятью и т.д.; с помощью драйверов возможно подключение к компьютеру новых устройств или нестандартное использование имеющихся устройств.

Идентификатор.

Символическое имя переменной, которое идентифицирует её в программе.

Инструментальные программные средства.

Программы, используемые в ходе разработки, корректировки или развития других программ: редакторы, отладчики, вспомогательные системные программы, графические пакеты и др. По назначению близки системам программирования.

Интегральная схема.

Реализация электронной схемы, выполняющей некоторую функцию, в виде единого полупроводникового кристалла, в котором изготовлены все компоненты, необходимые для осуществления этой функции.

Интегрированные пакеты программ.

Пакеты программ, выполняющие ряд функций, для которых ранее создавались специализированные программы — в частности, текстовые редакторы, электронные таблицы, системы управления базами данных, программы построения графиков и диаграмм.

Интернет.

Гигантская всемирная компьютерная сеть, объединяющая десятки тысяч сетей всего мира. Её назначение — обеспечить любому желающему постоянный доступ к любой информации. Интернет предлагает практически неограниченные информационные ресурсы, полезные сведения, учёбу, развлечения, возможность общения с компетентными людьми, услуги удалённого доступа, передачи файлов, электронной почты и многое другое. Интернет обеспечивает принципиально новый способ общения людей, не имеющий аналогов в мире.

Интерпретатор.

Разновидность транслятора. Переводит и выполняет программу с языка высокого уровня в машинный код строка за строкой.

Интерфейс.

Электронная схема сопряжения двух устройств, обменивающихся информацией.

Информатизация общества.

Организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления организаций, общественных объединений на основе формирования и использования информационных ресурсов.

Информатика.

Дисциплина, изучающая структуру и общие свойства информации, а также закономерности и методы её создания, хранения, поиска, преобразования, передачи и использования в различных сферах человеческой деятельности. За понятием "информатика" закреплены 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. Информатика в самом своем существе базируется на компьютерной технике.

Информационная технология.

Совокупность методов и устройств, используемых людьми для обработки информации. Охватывает всю вычислительную технику, технику связи и, отчасти, — бытовую электронику, телевизионное и радиовещание.

Информационно-поисковая система (ИПС).

Система, выполняющая функции хранения большого объёма информации, быстрого поиска требуемой информации, добавления, удаления и изменения хранимой информации, вывода её в удобном для человека виде.

Информация.

Сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые воспринимают информационные системы (живые организмы, управляющие машины и др.) в процессе жизнедеятельности и работы. Применительно к обработке данных на компьютерах — произвольная последовательность символов, несущих смысловую нагрузку.

Искусственный интеллект (ИИ).

Дисциплина, изучающая возможность создания программ для решения задач, которые требуют определённых интеллектуальных усилий при выполнении их человеком. Примерами областей использования ИИ являются: игры, логический вывод, обучение, понимание естественных языков, формирование планов, понимание речи, доказательство теорем и визуальное восприятие.

Исполнитель алгоритма.

Человек или автомат (в частности, процессор компьютера), умеющий выполнять определённый набор действий. Исполнителя характеризуют среда, элементарные действия, система команд, отказы.

Итерационный цикл.

Вид цикла, для которого число повторений операторов тела цикла заранее неизвестно. На каждом шаге вычислений происходит последовательное приближение и проверка условия достижения искомого результата. Выход из цикла осуществляется в случае выполнения заданного условия.

Каталог (директория, папка).

Оглавление файлов. Доступен пользователю через командный язык операционной системы. Его можно просматривать, переименовывать зарегистрированные в нём файлы, переносить их содержимое на новое место и удалять. Часто имеет иерархическую структуру.

Клавиатура компьютера.

Устройство для ввода информации в компьютер и подачи управляющих сигналов. Содержит стандартный набор клавиш печатающей машинки и некоторые дополнительные клавиши — управляющую клавишу, функциональные клавиши, клавиши управления курсором и малую цифровую клавиатуру.

Клиент (рабочая станция).

Любой компьютер, имеющий доступ к услугам сервера. Клиентом также называют прикладную программу, которая от имени пользователя получает услуги сервера. См. [Сервер](#).

Ключевое слово.

Слово языка программирования, имеющее определённый смысл для транслятора. Его нельзя использовать для других целей, например, в качестве имени переменной.

Команда.

Описание элементарной операции, которую должен выполнить компьютер. Обычно содержит код выполняемой операции, указания по определению операндов (или их адресов), указания по размещению получаемого результата. Последовательность команд образует программу.

Компакт-диск (CD—ROM).

Постоянное ЗУ, выполненное с использованием специальной оптической технологии. В ряду запоминающих устройств занимает место между флоппи- и жёстким дисками, являясь одновременно и мобильным и очень ёмким.

Компилятор.

Разновидность транслятора. Читает всю программу целиком, делает её перевод и создаёт законченный вариант программы на машинном языке, который затем и выполняется.

Компьютер.

Программируемое электронное устройство, способное обрабатывать данные и производить вычисления, а также выполнять другие задачи манипулирования символами. Основу компьютеров образует аппаратура (HardWare), построенная, в основном, с использованием электронных и электромеханических элементов и устройств. Принцип действия компьютеров состоит в выполнении программ (SoftWare) — заранее заданных, чётко определённых последовательностей арифметических, логических и других операций.

Компьютеризация.

Задачи массового внедрения компьютеров во все области жизни, стоящие перед странами как необходимое важное условие их прогресса и развития, а также последствия, которые будут вызваны этим массовым внедрением компьютеров.

Цель компьютеризации — улучшение качества жизни людей за счёт увеличения производительности и облегчения условий их труда.

Контроллер.

Устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

Курсор.

Светящийся символ на экране дисплея, указывающий позицию, на которой будет отображаться следующий вводимый с клавиатуры знак.

Кэш.

См. [Сверхоперативная память](#).

Логический тип.

Тип данных, представляемый значениями "истина" или "ложь" ("да" или "нет"). Иногда также называется булевым в честь английского математика XIX века Джорджа Буля.

Логический элемент (вентиль).

Часть электронной логической схемы, выполняющая элементарную логическую функцию.

Логическое высказывание.

Любое предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Локальная сеть (ЛВС).

См. [Сеть компьютерная](#).

Лэптоп (наколенник).

Портативный компьютер, по своим размерам близкий к портфелю. По быстродействию и памяти примерно соответствует настольным персональным компьютерам.

Манипуляторы (джойстик, мышь, трекболл и др.).

Специальные устройства для управления курсором.

Массив.

Последовательность однотипных элементов, число которых фиксировано и которым присвоено одно имя. Компьютерный эквивалент таблицы. Положение элемента в массиве однозначно определяется его индексами.

Математическая модель.

Система математических соотношений — формул, уравнений, неравенств и т.д., отражающих существенные свойства объекта.

Машинный язык.

Совокупность машинных команд компьютера, отличающаяся количеством адресов в команде, назначением информации, задаваемой в адресах, набором операций, которые может выполнить машина, и др.

Меню.

Выведенный на экран дисплея список различных вариантов работы компьютера, по которому можно сделать конкретный выбор.

Микропроцессор.

Процессор, выполненный в виде интегральной схемы. Состоит из цепей управления, регистров, сумматоров, счётчиков команд, очень быстрой памяти малого объёма.

Микрокомпьютер.

Компьютер, в котором в качестве управляющего и арифметического устройства используется микропроцессор.

Модем.

Устройство, обеспечивающее преобразование цифровых сигналов компьютера в переменный ток частоты звукового диапазона (модуляцию), а также обратное преобразование (демодуляцию). Используется для соединения компьютера с другими компьютерными системами через телефонную сеть.

Монитор.

См. [Дисплей](#).

Мультимедиа.

Собирательное понятие для различных компьютерных технологий, при которых используется несколько информационных сред, таких, как графика, текст, видео, фотография, движущиеся образы (анимация), звуковые эффекты, высококачественное звуковое сопровождение. *Мультимедиа-компьютер* — это компьютер, снабжённый аппаратными и программными средствами, реализующими технологию мультимедиа.

Мышь.

Устройство управления курсором. Имеет вид небольшой коробки, уместяющейся на ладони. Связана с компьютером кабелем. Её движения трансформируются в перемещения курсора по экрану дисплея.

Накопитель на жёстких магнитных дисках (винчестерский накопитель).

Наиболее массовое запоминающее устройство большой ёмкости, в котором носителями информации являются круглые алюминиевые пластины — платтеры, обе поверхности которых покрыты слоем магнитного материала. Используется для постоянного хранения больших объёмов информации.

Ноутбук (блокнот).

Портативный компьютер, по своим размерам близкий к книге крупного формата. Помещается в портфель-дипломат. Обычно комплектуется модемом и снабжается приводом CD—ROM.

Оболочки.

Программы, создаваемые для упрощения работы со сложными программными системами, такими, например, как операционная система DOS. Они преобразуют неудобный командный пользовательский интерфейс в дружелюбный графический интерфейс или интерфейс типа "меню". Оболочки предоставляют пользователю удобный доступ к файлам и обширные сервисные услуги.

Обработка информации.

В информатике — любое преобразование информации из одного вида в другой, производимое по строгим формальным правилам.

Оперативная память (ОЗУ).

Быстрое запоминающее устройство не очень большого объёма, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

Оператор.

Фраза алгоритмического языка, определяющая некоторый законченный этап обработки данных. В состав операторов входят ключевые слова, данные, выражения и др.

Операционная система.

Комплекс взаимосвязанных программ, предназначенных для автоматизации планирования и организации процесса обработки программ, ввода-вывода и управления данными, распределения ресурсов, подготовки и отладки программ, других вспомогательных операций обслуживания. Важнейшая часть программного обеспечения.

Описание.

Раздел программы, идентифицирующий структуры данных, которыми должна манипулировать программа, и описывающий их типы.

Основание системы счисления.

Количество различных цифр, используемых для изображения чисел в данной системе счисления.

Отладка (англ. *debugging*).

Этап компьютерного решения задачи, при котором происходит устранение явных ошибок в программе. Часто производится с использованием специальных программных средств — отладчиков.

Отладчик (англ. *debugger*).

Программа, позволяющая исследовать внутреннее поведение разрабатываемой программы. Обеспечивает пошаговое исполнение программы с остановкой после каждой оператора, просмотр текущего значения переменной, нахождение значения любого выражения и др.

Пакеты прикладных программ (ППП).

Специальным образом организованные программные комплексы, рассчитанные на общее применение в определённой проблемной области и дополненные соответствующей технической документацией.

Палмтоп (наладонник).

Самый маленький современный персональный компьютер. Умещается на ладони. Магнитные диски в нём заменяет энергонезависимая электронная память. Нет и накопителей на дисках — обмен информацией с обычными компьютерами идет по линиям связи.

Первое поколение компьютерной техники.

Машины, созданные на рубеже 50-х годов. В схемах использовались электронные лампы. Набор команд небольшой, схема арифметико-логического устройства и устройства управления простая, программное обеспечение практически отсутствовало. Быстродействие 10 — 20 тысяч операций в секунду.

Переменная.

Величина, значение которой может меняться в процессе выполнения программы.

Персональный компьютер.

Микрокомпьютер универсального назначения, рассчитанный на одного пользователя и управляемый одним человеком.

Подпрограмма.

Самостоятельная часть программы, которая создаётся независимо от других частей и затем вызывается по имени. Когда имя подпрограммы используется в качестве оператора программы, выполняется вся группа операторов, представляющая тело подпрограммы.

Поколения компьютеров.

Условная, нестрогая классификация вычислительных систем по степени развития аппаратных и программных средств, а также способов общения с ними.

Порты устройств.

Электронные схемы, содержащие один или несколько регистров ввода-вывода и позволяющие подключать периферийные устройства компьютера к внешним шинам микропроцессора. Последовательный порт обменивается данными с процессором побайтно, а с внешними устройствами — побитно. Параллельный порт получает и посылает данные побайтно.

Постоянная память (ПЗУ).

Энергонезависимое запоминающее устройство, изготовленное в виде микросхемы. Используется для хранения данных, не требующих изменения. Содержание памяти специальным образом "зашивается" в ПЗУ при изготовлении. В ПЗУ находятся программа управления работой самого процессора, программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.

Прикладная программа.

Любая конкретная программа, способствующая решению какой-либо задачи в пределах данной проблемной области.

Принтер.

Печатающее устройство. Преобразует закодированную информацию, выходящую из процессора, в форму, удобную для чтения на бумаге.

Принцип открытой архитектуры.

1. Регламентируются и стандартизируются только описание принципа действия компьютера и его конфигурация (определённая совокупность аппаратных средств и соединений между ними). Таким образом, компьютер можно собирать из отдельных узлов и деталей, разработанных и изготовленных независимыми фирмами-изготовителями.
2. Компьютер легко расширяется и модернизируется за счёт наличия внутренних расширительных гнезд, в которые пользователь может вставлять разнообразные устройства, удовлетворяющие заданному стандарту, и тем самым устанавливать конфигурацию своей машины в соответствии со своими личными предпочтениями.

Принципы фон-Неймана.

1. *Принцип программного управления.* Программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определённой последовательности.
2. *Принцип адресности.* Основная память состоит из перенумерованных ячеек; процессору времени доступна любая ячейка.
3. *Принцип однородности памяти.* Программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

Программное обеспечение (SoftWare).

Совокупность программ, выполняемых компьютером, а также вся область деятельности по проектированию и разработке программ.

Прокрутка.

Имитация программистом за столом выполнения программы на конкретном наборе тестовых данных.

Протокол коммуникации.

Согласованный набор конкретных правил обмена информацией между разными устройствами передачи данных. Имеются протоколы для скорости передачи, форматов данных, контроля ошибок и др.

Псевдокод.

Система обозначений и правил, предназначенная для единообразной записи алгоритмов. Занимает промежуточное место между естественным и формальным языками.

Регистр.

Специальная запоминающая ячейка, выполняющая функции кратковременного хранения числа или команды и выполнения над ними некоторых операций. Отличается от ячейки памяти тем, что может не только хранить двоичный код, но и преобразовывать его.

Регистр команд.

Регистр УУ для хранения кода команды на период времени, необходимый для её выполнения.

Сверхоперативная память.

Очень быстрое ЗУ малого объёма. Используется для компенсации разницы в скорости обработки информации процессором и несколько менее быстродействующей оперативной памятью.

Семантика.

Система правил истолкования отдельных языковых конструкций. Определяет смысловое значение предложений языка. Устанавливает, какие последовательности действий описываются теми или иными фразами языка и какой алгоритм определён данным текстом на алгоритмическом языке.

Сервер.

Высокопроизводительный компьютер с большим объёмом внешней памяти, который обеспечивает обслуживание других компьютеров путем управления распределением дорогостоящих ресурсов совместного пользования (программ, данных и периферийного оборудования). См. также [Клиент](#).

Сеть компьютерная.

Совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети. По степени географического распространения сети делятся на локальные, городские, корпоративные, глобальные и др.

Локальная сеть (ЛВС) – связывает ряд компьютеров в зоне, ограниченной пределами одной комнаты, здания или предприятия.

Глобальная сеть (ГВС) — соединяет компьютеры, удалённые географически на большие расстояния друг от друга. Отличается от локальной сети более протяженными коммуникациями (спутниковыми, кабельными и др.).

Городская сеть — обслуживает информационные потребности большого города.

Синтаксис.

Набор правил построения фраз языка, позволяющий определить, какие комбинации символов являются осмысленными предложениями в этом языке.

Система команд.

Совокупность операций, выполняемых некоторым компьютером.

Система программирования.

Система для разработки новых программ на конкретном языке программирования. Предоставляет пользователю мощные и удобные средства разработки программ: транслятор, редактор текстов программ, библиотеки стандартных программ, отладчик и др.

Система счисления.

Совокупность приемов и правил, по которым записываются и читаются числа.

Система телеконференций.

Основанная на использовании компьютерной техники система, позволяющая пользователям, несмотря на их взаимную удалённость в пространстве, а иногда, и во времени, участвовать в совместных мероприятиях, таких, как организация и управление сложными проектами.

Система управления базами данных (СУБД).

Система программного обеспечения, позволяющая обрабатывать обращения к базе данных, поступающие от прикладных программ конечных пользователей.

Системные программы.

Программы общего пользования, выполняемые вместе с прикладными и служащие для управления ресурсами компьютера — центральным процессором, памятью, вводом-выводом.

Системы автоматизированного проектирования (САПР).

Комплексные программно-технические системы, предназначенные для выполнения проектных работ с применением математических методов. Широко используются в архитектуре, электронике, механике и др. В качестве входной информации в САПР используются технические знания специалистов, которые вводят проектные требования, уточняют результаты, проверяют полученную конструкцию, изменяют

ее и т.д. В САПР накапливается информация, поступающая из библиотек стандартов (данные о типовых элементах конструкций, их размерах, стоимости и др.).

Системы деловой графики.

Программные системы, позволяющие создавать различные виды графиков и диаграмм: гистограммы, круговые и секторные диаграммы и т.д.

Системы научной и инженерной графики.

Программные системы, позволяющие в цвете и в заданном масштабе отображать на экране графики двумерных и трехмерных функций, заданных в табличном или аналитическом виде, системы изолиний, в том числе и нанесенные на поверхность объекта, сечения, проекции, карты и др.

Сканер.

Устройство для ввода в компьютер документов — текстов, чертежей, графиков, рисунков, фотографий. Создает оцифрованное изображение документа и помещает его в память компьютера.

Сопровождение программ.

Работы, связанные с обслуживанием программ в процессе их эксплуатации.

Стример.

Устройство для резервного копирования больших объемов информации. В качестве носителя применяются кассеты с магнитной лентой ёмкостью 1—2 Гбайта и больше.

Структурное программирование.

Метод разработки программ, в частности, требующий разбиения программы на небольшие независимые части (модули). Обеспечивает возможность проведения строгого доказательства правильности программ, повышает уверенность в правильности конечной программы.

Сумматор.

Электронная логическая схема, выполняющая суммирование двоичных чисел.

Суперкомпьютер.

Очень мощный компьютер с производительностью свыше 100 мегафлопов (1 мегафлоп — миллион операций с плавающей точкой в секунду). Представляет собой многопроцессорный и (или) многомашинный комплекс, работающий на общую память и общее поле внешних устройств. Архитектура основана на идеях параллелизма и конвейеризации вычислений.

Схема алгоритма (блок-схема).

Графическое представление алгоритма в виде последовательности блоков, соединённых стрелками.

Счётчик команд.

Регистр УУ, содержимое которого соответствует адресу очередной выполняемой команды; служит для автоматической выборки команд программы из последовательных ячеек памяти.

Таблица истинности.

Табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

Текстовый редактор.

Программа для ввода и изменения текстовых данных (документов, книг, программ, ...). Обеспечивает редактирование строк текста, контекстный поиск и замену частей текста, автоматическую нумерацию страниц, обработку и нумерацию сносок, выравнивание краёв абзаца, проверку правописания слов и подбор синонимов, построение оглавлений, распечатку текста на принтере и др.

Тест.

Некоторая совокупность данных для программы, а также точное описание всех результатов, которые должна выработать программа на этих данных, в том виде, как эти результаты должны быть выданы программой.

Тестирование.

Этап решения задачи на компьютере, в процессе которого проверяется работоспособность программы, не содержащей явных ошибок.

Тип данных.

Понятие языка программирования, определяющее структуру констант, переменных и других элементов данных, разрешенные их значения и операции, которые можно над ними выполнять.

Топология.

Раздел математики, изучающий свойства фигур, не изменяющиеся при любых деформациях, производимых без разрывов и склеиваний. Понятие топологии широко используется при создании компьютерных сетей.

Топология компьютерной сети.

Логический и физический способ соединения компьютеров, кабелей и других компонентов, в целом составляющих сеть. Топология характеризует свойства сетей, не зависящие от их размеров. При этом не учитывается производительность и принцип работы этих объектов, их типы, длины каналов, хотя при проектировании эти факторы очень важны. Наиболее распространенные виды топологий: линейная, кольцевая, древовидная, звездообразная, ячеистая, полносвязная.

Транслятор.

Программа-переводчик. Преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Трекболл.

Устройство управления курсором. Небольшая коробка с шариком, встроенным в верхнюю часть её корпуса. Пользователь рукой вращает шарик и перемещает, соответственно, курсор.

Третье поколение компьютерной техники.

Семейства программно совместимых машин с развитыми операционными системами. Обеспечивают мультипрограммирование. Быстродействие внутри семейства от нескольких десятков тысяч до миллионов операций в секунду.

Ёмкость оперативной памяти — нескольких сотен тысяч слов. Элементная база — интегральные схемы.

Триггер.

Электронная схема, широко применяемая в регистрах компьютера для надёжного запоминания одного бита информации. Имеет два устойчивых состояния, которые соответствуют двоичной "1" и двоичному "0".

Упаковщики (архиваторы).

Программы, позволяющие записывать информацию на дисках более плотно, а также объединять копии нескольких файлов в один архивный файл.

Устройство управления (УУ).

Часть процессора, выполняющая функции управления устройствами компьютера.

Файл.

Именованная совокупность любых данных, размещенная на внешнем запоминающем устройстве и хранимая, пересылаемая и обрабатываемая как единое целое. Файл может содержать программу, числовые данные, текст, закодированное изображение и др. Имя файла регистрируется в каталоге.

Цикл.

Приём в программировании, позволяющий многократно повторять одну и ту же последовательность команд (операторов).

Четвёртое поколение компьютерной техники.

Теперьшнее поколение машин, разработанных после 1970 года. Эти компьютеры проектировались в расчёте на эффективное использование современных высокоуровневых языков и упрощение процесса программирования для конечного пользователя. Элементная база — интегральные схемы. Ёмкость ОЗУ — десятки Мегабайт. Машины этого поколения представляют собой персональные компьютеры, либо многопроцессорные и (или) многомашинные комплексы, работающие на общую память и общее поле внешних устройств. Быстродействие до нескольких десятков - сотен миллионов операций в секунду.

Чувствительный экран.

Позволяет осуществлять общение с компьютером путем прикосновения пальцем к определённому месту экрана монитора.

Штриховой код (бар-код).

Серия широких и узких линий, в которых зашифрован номер торгового изделия. Имеет большое распространение в организации компьютерного обслуживания торговых предприятий.

Экспертная система.

Комплекс компьютерного программного обеспечения, помогающий человеку принимать обоснованные решения. Использует информацию, полученную заранее от экспертов — людей, которые в какой-либо области являются лучшими специалистами. Хранит знания об определённой предметной области. Обладает комплексом логических средств для выведения новых знаний, выявления закономерностей, обнаружения противоречий и др.

Электронный офис.

Система автоматизации работы учреждения, основанная на использовании компьютерной техники.

Электронная почта.

Система пересылки сообщений между пользователями вычислительных систем, в которой компьютер берёт на себя все функции по хранению и пересылке сообщений.

Электронная таблица.

Программа, обрабатывающая таблицы, состоящие из строк и граф, на пересечении которых располагаются клетки. В клетках содержится числовая информация, формулы или текст. Значение в числовой клетке таблицы либо записано, либо рассчитано по формуле. В формуле могут присутствовать обращения к другим клеткам.

Язык ассемблера.

Система обозначений, используемая для представления в удобочитаемой форме программ, записанных в машинном коде. Перевод программы с языка ассемблера на машинный язык осуществляется специальной программой, которая называется *ассемблером* и является, по сути, простейшим транслятором.

Язык высокого уровня.

Язык программирования, более близкий к естественному языку, чем машинный код или язык ассемблера. Каждый оператор в нём соответствует нескольким командам машинного кода или языка ассемблера.

1. Примерные экзаменационные билеты по информатике для проведения устной итоговой аттестации выпускников XI классов общеобразовательных учреждений в 2000/2001 учебном году.

Билеты разработаны Министерством общего и профессионального образования РФ. Предназначены для общеобразовательных школ, оснащенных вычислительной техникой, в которых после прохождения базового курса информатики в 7-9 классах продолжается изучение курса в старших классах (10-11 классы). Если школа не располагает необходимыми условиями для выполнения практических заданий по информационным технологиям, такие задания могут быть заменены на другие задания, например, по алгоритмизации и программированию, или задачи.

Билет № 1

- 1. Информация и информационные процессы в природе, обществе, технике. Информационная деятельность человека. Привести примеры.**
- 2. Объектно-ориентированное программирование. Объекты: свойства и методы. Классы объектов.**
- 3. Практическое задание на проведение расчетов с помощью электронной таблицы.**

Билет № 2

- 1. Информационные процессы в управлении. Обратная связь. Показать на примере.**
- 2. Строковые величины. Строковые функции и выражения.**
- 3. Задача на определение результата выполнения алгоритма по его блок-схеме или записи на алгоритмическом языке.**

Билет № 3

- 1. Язык и информация. Естественные и формальные языки. Алгоритмическое программирование. Основные способы организации действий в алгоритмах. Практическое задание на построение графика функции с помощью электронной таблицы.**

Билет № 4

- 1. Двоичная система счисления. Запись чисел в двоичной системе счисления.**
- 2. Компьютер. Его состав: магистрально-модульное построение компьютера.**
- 3. Задание на разработку фрагмента программы тестирования знаний.**

Билет № 5

- 1. Кодирование информации. Способы кодирования. Привести примеры.**
- 2. Основные характеристики компьютера (разрядность, объем оперативной и внешней памяти и др.).**
- 3. Практическое задание на создание, преобразование, сохранение и печать рисунка с помощью графического редактора.**

Билет № 6

- 1. Качественные и количественные характеристики информации. Свойства информации (новизна, актуальность, достоверность и др.). Единицы измерения количества информации.**
- 2. Внешняя память компьютера. Различные виды носителей информации, их характеристики.**
- 3. Практическое задание на работу с файлами (копирование, переименование, удаление).**

Билет № 7

- 1. Функциональная схема компьютера. Основные устройства компьютера, их назначение и взаимосвязь.**
- 2. Способы записи алгоритмов (описательный, графический, на алгоритмическом языке, на языке программирования).**
- 3. Практическое задание на поиск файлов.**

Билет № 8

- 1. Программное управление работой компьютера. Программное обеспечение компьютера.**
- 2. Основные типы и способы организации данных (переменные, массивы, списки).**
- 3. Практическое задание на работу с диском (форматирование, создание системной дискеты).**

Билет № 9

- 1. Папки и файлы (тип файла, имя файла). Файловая система. Основные операции с файлами в операционной системе.**
- 2. Логическое сложение. Таблица истинности.**
- 3. Практическое задание на тестирование и <лечение> гибкого диска от вирусов.**

Билет № 10

- 1. Правовая охрана программ и данных. Защита информации.**
- 2. Основные логические устройства компьютера (сумматор, регистр).**
- 3. Практическое задание на создание, редактирование и распечатку текста с помощью текстового редактора.**

Билет № 11

- 1. Этапы решения задачи с помощью компьютера (построение модели — формализация модели — построение компьютерной модели — проведение компьютерного эксперимента — интерпретация результата).**
- 2. Моделирование как метод научного познания. Модели материальные и информационные.**
- 3. Практическое задание на поиск информации в базе данных по заданным параметрам.**

Билет № 12

1. Формализация моделей. Привести пример формализации (например, преобразование описательной модели в математическую).
2. Мультимедиа технология.
3. Практическое задание на сортировку информации в базе данных по заданным параметрам.

Билет № 13

1. Описание состояния объекта и описание изменения состояния объекта с помощью статических и динамических информационных моделей. Примеры из различных предметных областей.
2. Массивы и алгоритмы их обработки.
3. Задача на перевод числа, записанного в десятичной системе счисления, в двоичную систему, восьмеричную и шестнадцатеричную системы.

Билет № 14

1. Алгоритм. Свойства алгоритма. Возможность автоматизации деятельности человека.
2. Операционная система компьютера (назначение, состав, загрузка). Графический интерфейс.
3. Задание на разработку программы по подсчету количества появлений конкретного символа в заданном фрагменте текста.

Билет № 15

1. Алгоритмическая структура <ветвление>. Команды ветвления. Привести пример.
2. Представление и кодирование информации с помощью знаковых систем. Алфавитный подход к определению количества информации.
3. Задача на определение истинности составного высказывания.

Билет № 16

1. Алгоритмическая структура <цикл>. Команды повторения. Привести пример.
2. Выполнение арифметических операции в двоичной системе счисления.
3. Задача на определение количества информации с последующим преобразованием единиц измерения.

Билет № 17

1. Пример разработки алгоритма методом последовательной детализации. Вспомогательные алгоритмы.
2. Информационное моделирование. Основные типы информационных моделей.
3. Задача на сложение и вычитание двоичных чисел.

Билет № 18

1. Основы языка программирования (алфавит, операторы, типы данных и т.д.).
2. Основы языка разметки гипертекста (HTML).
3. Практическое задание на организацию запроса при поиске информации в Интернете.

Билет № 19

- 1. Текстовый редактор. Назначение и основные функции.**
- 2. Двоичное кодирование текстовой информации. Различные кодировки кириллицы.**
- 3. Практическое задание на инсталляцию программного продукта.**

Билет № 20

- 1. Графический редактор. Назначение и основные функции.**
- 2. Логическое умножение. Таблица истинности.**
- 3. Привести пример адреса электронной почты и объяснить его формат.**

Билет № 21

- 1. Электронные таблицы. Назначение и основные функции.**
- 2. Адресация в Интернете: доменная система имен и IP-адреса.**
- 3. Задание на разработку программы поиска максимального элемента в массиве.**

Билет № 22

- 1. Базы данных. Назначение и основные функции.**
- 2. Компьютерные вирусы: способы распространения, защита от вирусов.**
- 3. Практическое задание на разработку Web-страницы.**

Билет № 23

- 1. Информационные ресурсы сети Интернет: электронная почта, телеконференции, файловые архивы. Всемирная паутина.**
- 2. Информация. Вероятностный подход к измерению количества информации.**
- 3. Задача на построение блок-схемы алгоритма, записанного на естественном языке.**

Билет № 24

- 1. Гипертекст. Технология WWW (World Wide Web — Всемирная паутина).**
- 2. Визуальное объектно-ориентированное программирование. Графический интерфейс: форма и управляющие элементы.**
- 3. Практическое задание на определение информационной емкости носителей информации.**

Билет № 25

- 1. Основные этапы развития вычислительной техники. Информатизация общества.**
- 2. Локальные и глобальные компьютерные сети. Назначение сетей.**
- 3. Задание на разработку программы с использованием двумерного массива и вложенных циклов.**