

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЛЕСОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

**Л.Ю.Мельник**

ИНФОРМАТИКА  
**Лекция (основные понятия)**

ЕКАТЕРИНБУРГ, 2020

## Оглавление

<b>ВВЕДЕНИЕ</b> .....	6
<b>ИНФОРМАТИКА – ПРЕДМЕТ И ЗАДАЧИ</b> .....	7
<b>Структура информатики</b> .....	7
Задачи информатики: .....	8
<b>Основы информационной культуры</b> .....	9
Информатизация общества .....	9
<b>Справка о смене поколений ЭВМ</b> .....	9
<b>Роль информации в развитии общества</b> .....	11
Об информационной культуре.....	12
<b>ИНФОРМАЦИОННЫЙ ПОТЕНЦИАЛ ОБЩЕСТВА</b> .....	12
<b>Информационные продукты и услуги</b> .....	13
<b>ИЗМЕРЕНИЕ И ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ</b> .....	14
<b>Информация и ее свойства</b> .....	14
<b>МЕРЫ ИНФОРМАЦИИ</b> .....	15
Классификация мер.....	15
<b>Семантическая мера информации</b> .....	18
<b>Прагматическая мера информации</b> .....	19
<b>КАК ПЕРЕДАЁТСЯ ИНФОРМАЦИЯ?</b> .....	21
<b>Что можно делать с информацией?</b> .....	21
<b>Какими свойствами обладает информация?</b> .....	22
<b>КАЧЕСТВО ИНФОРМАЦИИ</b> .....	22
<b>ФОРМЫ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ</b> .....	24
<b>Понятие о единицах измерения информации в ЭВМ:</b> .....	27
<b>ЧТО ТАКОЕ КОМПЬЮТЕР?</b> .....	28
<b>Как устроен компьютер?</b> .....	29
На каких принципах построены компьютеры?.....	31
<b>Что такое команда?</b> .....	32
Как выполняется команда? .....	33

Что такое архитектура и структура компьютера?.....	34
Что такое центральный процессор?.....	36
Как устроена память? .....	37
Какие устройства образуют внутреннюю память? .....	38
Оперативная память .....	38
Кэш-память .....	39
Специальная память .....	39
<b>КАКИЕ УСТРОЙСТВА ОБРАЗУЮТ ВНЕШНЮЮ ПАМЯТЬ?.....</b>	<b>41</b>
Накопители на гибких магнитных дисках.....	41
Накопители на жестких магнитных дисках .....	42
Накопители на компакт-дисках .....	42
Записывающие оптические и магнитооптические накопители .....	44
Накопители на магнитной ленте (стримеры) и накопители на сменных дисках .....	45
Что такое аудиоадаптер?.....	45
Что такое видеоадаптер и графический акселератор? .....	46
Что такое клавиатура? .....	47
Что такое видеосистема компьютера? .....	48
Монитор на базе электронно-лучевой трубки .....	49
Жидкокристаллические мониторы.....	50
Сенсорный экран.....	51
<b>2Что такое принтер, плоттер, сканер? .....</b>	<b>51</b>
Что такое модем и факс-модем? .....	53
Что такое манипуляторы? .....	54
<b>КАК УСТРОЕН КОМПЬЮТЕР?.....</b>	<b>55</b>
Какие основные блоки входят в состав компьютера? .....	57
Что собой представляет системная плата? .....	58
<b>ИНФОРМАЦИЯ И ЕЕ СВОЙСТВА.....</b>	<b>58</b>
Основные определения.....	58
Классификация информации .....	60
Количественное измерение информации.....	62
Кодирование различных типов информации .....	63

ПРИМЕРЫ ТЕСТОВ.....	65
ОСНОВНЫЕ ПОНЯТИЯ СИСТЕМ СЧИСЛЕНИЯ .....	67
Виды систем счисления .....	68
Правила перевода чисел из одной системы счисления в другую .....	73
АЛГЕБРА ЛОГИКИ И ДОКАЗАТЕЛЬСТВО.....	77
Высказывания, операции над высказываниями .....	77
Логические законы .....	83
Предикаты и кванторы .....	84
Логические и битовые операции .....	87
КЛАССИФИКАЦИЯ МОДЕЛЕЙ ДАННЫХ.....	88
Инфологические модели.....	89
Даталогические модели .....	90
Иерархическая модель.....	90
Сетевая модель.....	92
Реляционная модель.....	93
Что понимается под верификацией модели.....	94
ЭТАПЫ ИНФОРМАЦИОННОГО МОДЕЛИРОВАНИЯ.....	95
1 Этап: .....	95
2 Этап: .....	96
3 Этап: .....	96
4 Этап: .....	96
ОСНОВНЫЕ ЭТАПЫ РЕШЕНИЯ ЗАДАЧ НА КОМПЬЮТЕРЕ.....	96
Порядок решения задач на компьютере (пример): .....	97
СПОСОБЫ ЗАПИСИ АЛГОРИТМОВ.....	98
Основные свойства алгоритма .....	99
Основные алгоритмические структуры. ....	100

СОСТАВ СРЕДСТВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ .....	104
ТРАНЛЯЦИЯ, КОМПИЛЯЦИЯ И ИНТЕРПРЕТАЦИЯ .....	104
ЭВОЛЮЦИЯ И КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ. ОСНОВНЫЕ ПОНЯТИЯ ЯЗЫКОВ.....	107
ИНСТРУМЕНТАЛЬНЫЕ СИСТЕМЫ ПРОГРАММИРОВАНИЯ .....	111
КЛАССИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	111
Краткий обзор прикладного программного обеспечения .....	113
Вспомогательные программы .....	114
Операционные системы .....	115
Операционная система Windows .....	118
ЭВОЛЮЦИЯ И КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ .....	123
<b>Структурное программирование.</b> .....	<b>123</b>
Программирования сверху – вниз. ....	124
Программирования снизу – вверх. ....	124
Способы реализации основного принципа структурного программирования: .....	125
Достоинства структурного программирования: .....	125
<b>Модульный принцип программирования</b> .....	<b>126</b>
<b>Объектно-ориентированное программирование.</b> .....	<b>126</b>
Базовые принципы ООП .....	128
<b>Типы</b> .....	<b>128</b>
Достоинства ООП.....	129
ОСНОВНЫЕ ХАРАКТЕРИСТИКИ И КЛАССИФИКАЦИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ .....	130
Топология сетей .....	131
Сетевое оборудование .....	134
История развития Internet .....	137
Структура и принципы работы Интернет.....	138
Протоколы передачи данных .....	140
Подключение к Интернет .....	142
WWW и HTML .....	143
Браузеры .....	145

Поиск информации в Интернет.....	145
Электронная почта .....	148
FTP.....	151

## Введение

Информатика и информационные технологии - необходимый атрибут профессиональной пригодности в обществе. Претендент на престижную работу должен обладать не только профессиональными знаниями и владеть иностранным языком, но и свободно разбираться в информационной технологии работы на персональном компьютере (ПК), что и как раз дает изучение информатики.

Наступило время, когда профессионал - юрист, инженер, экономист, социолог, журналист - уже с трудом справляется с потоками информации. Специалисту, чтобы на должном уровне выполнять свои обязанности, необходимы инструментальный и методология его применения для обработки информации. Это сравнимо с использованием средств передвижения: теоретически человек может пешком преодолеть любое расстояние, но современный темп жизни просто невыносим без применения автомобиля, поезда, самолета и т.д. То же самое происходит и в области, связанной с обработкой информации: теоретически человек сам может переработать любую информацию без компьютера, но сделает это эффективнее, если овладеет знаниями и умениями, которыми располагает информатика.

Информатика служит прежде всего для формирования определенного мировоззрения в информационной сфере и освоения информационной культуры, т.е. умения целенаправленно работать с информацией, профессионально используя ее для получения, обработки и передачи в компьютерную информационную технологию и соответствующие ей технические и программные средства.

Цель предмета:

- Дать целостное представление об информатике и ее роли в развитии общества;
- Раскрыть суть и возможности технических и программных средств информатики;
- Сформировать понимание - с какой целью и каким образом можно использовать информационные системы и технологии.

Практикум по информатике - необходимое, неотъемлемое дополнение предмета.

Цель практикума - научить пользоваться программным инструментарием компьютерной информационной технологии для работы:

- на локальном компьютере и при подключении его к сети;
- с документами и текстами;
- с данными, представленными в табличной форме;
- с базами данных.

## Информатика – предмет и задачи.

Термин *информатика* возник в 60 –х гг. во Франции для названия области, занимающейся автоматизированной обработкой информации с помощью ЭВМ. Французский термин образован слияние двух слов информация и автоматика, т.е. информационная автоматика.

Выделение информации как самостоятельной области человеческой деятельности связано с развитием компьютерной техники.

**Информатика** – это область человеческой деятельности, связанная с процессами преобразования информации с помощью компьютеров и их взаимодействием со средой применения.

**Кибернетика** – это наука об общих принципах управления в различных системах: технических, биологических, социальных и др.

Кибернетика существует независимо от наличия и отсутствия компьютеров.

Информатика появилась благодаря развитию компьютерной техники, базируется на ней. Кибернетика развивается сама по себе, строя различные модели упорядочения объектами, хотя и очень активно использует достижения компьютерной техники.

## Структура информатики.

Информатика в широком смысле представляет собой единство разнообразных отраслей науки, техники и производства, связанных с переработкой информации гл. образом с помощью компьютеров во всех сферах человеческой деятельности.

Информатику в узком смысле можно представить, как состоящую из трех взаимосвязанных частей – технических ср-в, программных средств и алгоритмических.



### Рис.1.1. Структура Информатики

Информатика как *отрасль народного хозяйства* состоит из однородной совокупности предприятий разных форм хозяйствования, где занимаются производством компьютерной техники, программных продуктов и переработкой информации. От нее во многом зависит рост производительности труда в других отраслях народного хозяйства.

Информатика как *фундаментальная наука* занимается методологией создания информационного обеспечения процессов управления любыми объектами на базе компьютерных информационных систем. Цель фундаментальных исследований в информатике – получение обобщенных знаний о любых информационных системах, выявление общих закономерностей их построения и функционирования.

**Главная функция информатики** – разработка средств и методов преобразования информации и их использование в организации технологического процесса переработки информации.

Задачи информатики:

- исследование информационных процессов любой природы;
- разработка информационной техники и создание новейшей технологии переработки информации на базе полученных результатов исследования информационных процессов;

- решение научных и инженерных проблем эффективного использования компьютерной техники и технологии во всех сферах общественной жизни.

## Основы информационной культуры.

### Информатизация общества.

В истории развития цивилизации произошло несколько информационных революций - преобразований общественных отношений из-за кардинальных изменений в сфере обработки информации.

Первая революция связана с изобретением письменности, что привело к гигантскому качественному и количественному скачку. Появилась возможность передачи знаний от поколения к поколениям.

В т о р а я (середина XVI в.) вызвана изобретением книгопечатания, которое радикально изменило индустриальное общество, культуру, организацию деятельности.

Т р е т ь я (середина XIX в.) обусловлена изобретением электричества, благодаря которому появились телеграф, телефон, радио, позволяющие оперативно передавать и накапливать информацию в любом объеме.

Ч е т в е р т а я (70 -е гг. XX в.) связана с изобретением микропроцессорной технологии и появлением персонального компьютера. На микропроцессорах и интегральных схемах создаются компьютеры, компьютерные сети, системы передачи данных (информационные коммуникации). Этот период характеризуют три фундаментальные инновации:

- Переход от механических и электрических ср-в преобразования информации к электронным;
- Миниатюризация всех узлов;
- Создание программно - управляемых устройств и процессов.

### Справка о смене поколений ЭВМ.

1 -е поколение (начало 50 -х гг.). Элементная база - электронные лампы. ЭВМ отличались большими габаритами, большим потреблением энергии, малым быстродействием, низкой надежностью, программированием в кодах).

2 -е поколение (с конца 50 -х гг.). Элементная база - полупроводниковые элементы. Улучшились по сравнению с ЭВМ предыдущего поколения все тех-е хар-ки. Для программирования используются алгоритмические языки.

3 -поколение (начало 60 -х гг.). Элементная база - интегральные схемы, многослойный печатный монтаж. Резкое снижение габаритов ЭВМ, повышение их надежности, увеличение производительности.

4 -поколение (с середины 70 -х гг.). Элементная база - микропроцессоры, большие интегральные схемы. Улучшились тех. характеристики. Массовый выпуск перс-х компьютеров.

5 - поколение (с середины 80 -х гг.). Началась разработка интеллектуальных компьютеров, пока не увенчавшаяся успехом. Внедрение во все сферы компьютерных сетей, повсеместное применение компьютерных информационных технологий.

Последняя инф. революция выдвигает на первый план новую отрасль - *инф -ю индустрию*, связанную с производством тех. средств для производства новых знаний.

**Информационная технология (ИТ)**-процесс, использующий совокупность ср-в и методов обработки данных (первичной информации) для получения информации нового качества о состоянии объекта.

**Телекоммуникация** - дистанционная передача данных на базе компьютерных сетей и соврем. технических средств связи.

Бурное развитие компьютерной техники и информационных технологий послужило толчком к развитию общества, построенного на использовании различной информации и получившего название информационного общества (ИО).

В ИО изменится система ценностей, от человека потребуются способность к творчеству, возрастет спрос на знания.

**ИО** - общество, в котором большинство работающих занято производством, хранением, переработкой и реализацией информации, особенно высшей ее формы - знаний.

#### **Характерные черты информационного общества:**

- Решена проблема информационного кризиса;
- Обеспечен приоритет информации по сравнению с другими ресурсами;
- Гл. формой развития станет информационная экономика;
- В основу общества будут заложены использ-е знаний с помощью новейшей ИТ;
- ИТ приобретет глобальный хар-р: реализованы гуманистические принципы упр-я обществом и воздействия на окружающую среду.

Кроме положительного момента прогнозируются и опасные тенденции:

- Все большее влияние на общество средств массовой информации;
- ИТ могут разрушить частную жизнь людей и организаций;
- Существует проблема отбора качественной и достоверной информации;
- Многим людям будет тяжело адаптироваться в среде ИО.

Ближе всех на пути к ИО стоят страны с развитой информационной индустрией: США, Япония, Англия, Германия, З.Европа.

### Роль информации в развитии общества.

В ежедневно появляющемся потоке информации ориентироваться все трудней. Как результат - *информационный кризис (взрыв)*, который проявляется в следующем:

- Противоречие между мощными потоками информации и возможностью ее переработки (сначала сумма знаний менялась медленно, но уже с 1900 г. она удваивалась каждые 50 лет, к 1950 г.- каждые 10 лет, к 1970 г.- каждые 5 лет, с 1990 - ежегодно);
- Большое кол-во избыточной информации, которая затрудняет восприятие полезной;
- Экономические, политические и др. социальные барьеры распространения информации (секретность).

Внедрение ЭВМ послужило началом нового эволюционного процесса - *информатизации*.

**Информатизация общества** - организованный соц.-экон. и н.-т. прогресс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов гос. власти.

При компьютеризации общества осн. внимание уделяется развитию и внедрению технической базы компьютеров, обеспечивающих оперативное получение информации и ее накопление.

При информатизации общества основное внимание уделяется комплексу мер на полное использование достоверного знания во всех видах чел. деятельности.

ИО - более широкое понятие, чем КО.

Пример: в станкостроительной отрасли в 1990 г. было занято 330 тыс. человек, а к 2005 г. по прогнозам останется 14 тыс. человек. Это произойдет за счет массового сокращения людей на сборочных линиях, внедрения вместо них роботов и манипуляторов.

### Об информационной культуре.

В период перехода к информационному обществу необходимо подготовить человека к быстрому восприятию и обработке больших объемов информации, овладению их современными средствами, методами и технологией работы. Человек должен иметь определенный уровень культуры по обращению с информацией. Для отражения этого факта был введен термин *информационная культура (ИК)*.

**ИК** – умение целенаправленно работать с информацией и использовать для ее получения, обработки и передачи компьютерную ИТ, современные тех. ср-ва и методы.

ИК связана с социальной природой человека. Она является продуктом разнообразных творческих способностей человека и проявляется в следующих аспектах:

- В конкрет-х навыках по использованию тех. устройств ( от телефона до ПК и компьютерных сетей);
- В способности использовать в своей деятельности компьютерную ИТ;
- В умении извлекать и эффективно использовать информацию из различных источников;
- Во владении аналитической переработки информации;
- В знании особенностей информационных потоков в своей области деятельности.

В информационном обществе необходимо начать овладевать ИК с детства, сначала с помощью электронных игрушек, а затем привлекая ПК.

### Информационный потенциал общества.

#### **Информационные ресурсы.**

В ИО акцент внимания и значимости смещается с традиционных видов ресурсов на информационный ресурс (ИР).

ИР – отдельные документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, др. информационных системах).

ИР общества, если их понимать, как знания, отчуждены от людей, которые их накапливали, анализировали, обобщали. Эти знания материализовались в виде документов, баз данных, компьютерных программ, а также произведений искусства, литературы, науки.

ИР страны, региона должны рассматриваться как стратегические ресурсы, аналогичные запасам сырья, энергии и пр. ресурсам.

Развитие мировых ИР позволило:

- Превратить деятельность по оказанию информационных услуг в глобальную человеческую деятельность;
- Сформировать рынок информационных услуг;
- Образовать всевозможные базы данных ресурсов регионов и гос-в, к которым возможен сравнительно недорогой доступ;
- Повысить оперативность и обоснованность принимаемых решений в фирмах, банках, торговле и др. за счет своевременного использования необходимой информации.

### Информационные продукты и услуги.

ИО являются базой для создания информационных продуктов (ИП). ИП, являясь результатом интеллектуальной деятельности человека, должен быть зафиксирован в виде документов, книг и т.д.

**ИП** – совокупность данных, сформированная производителем для распространения в вещественной и невещественной форме.

ИП может распространяться такими же способами, как и любой другой материальный продукт, с помощью услуг.

**Услуга** – результат непроизводственной деятельности предприятия или лица, направленный на удовлетворение потребности человека или организации в использовании различных продуктов.

**Информационная услуга** – получение и предоставление в распоряжение пользователя ИП.

Библиотеки – место сосредоточения значительной части ИР страны. Основные виды ИО, оказываемых библиотечной сферой:

- предоставление полных текстов документов, а также справок по их описанию и местонахождению;
- выдача аналитической переработки информации (справки, указатели и т.д.);
- получение аналитической переработки информации (справки, указатели и т.д.);
- организация научно-технической пропаганды и рекламной деятельности (выставки, семинары, конференции);
- выдача результатов информационного исследования (отчеты, обзоры).

Информационные услуги возникают только при наличии баз данных в компьютерном и некомпьютерном варианте.

**База данных** – совокупность связанных данных, правила организации которых основаны на общих принципах описания, хранения и манипулирования данными.

БД являются источником и своего рода полуфабрикатом при подготовке ИУ соответствующими службами. БД существовали и до компьютерного периода в библиотеках, архивах и т.п.

БД разделяют на библиографические и неблиографические.

*Библиографические* БД содержат вторичную информационную о документах.

*Небиблиографические* БД имеют множество видов:

- Справочные (адреса, телефоны, расписания);
- Полного текста, содержащие первичную информационную (статьи, журналы, брошюры);
- Числовые, содержащие количественные характеристики;
- Текстово – числовые, содержащие описания объектов и их характеристики;
- Финансовые;
- Юридические, содержащие правовые документы по отраслям, регионам, странам.

*Ретроспективный поиск информации*- целенаправленный поиск информации в БД и пересылка рез-в по заявке пользователя.

*Дистанционный доступ(ДД)* к удаленным БД организуется в компьютерной сети в диалоговом режиме.

*Подготовка и оказание информационных услуг:*

- Связь (телефонная, телекоммуникационная);
- Обработка данных в вычислительных центрах;
- Программное обеспечение;
- Разработка информационных систем;
- Разработка информационных технологий.

## Измерение и представление информации.

Информацию можно оценивать, как на качественном, так и на количественном уровне, что для успешной обработки информации необходимо систематизировать, проводя ее классификацию и кодирование.

## Информация и ее свойства.

Термин **информация** происходит от латинского **informatio**, что означает разъяснение, осведомление, изложение. С позиции материалистической философии информация есть отражение реального мира с помощью сведений (сообщений).

**Сообщение** – это форма представления информации в виде речи, текста, изображения, цифровых данных, графиков, таблиц и т.п. В широком смысле информация – это общенаучное понятие, включающее в себя обмен сведениями между людьми, обмен сигналами между живой и неживой природой, людьми и устройствами.

**Информация** – сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний.

Наряду с информацией в информатике часто употребляется понятие *данные*. Данные могут рассматриваться как признаки или записанные наблюдения, которые по каким-то причинам не используются, а только хранятся. В том случае, когда появляется возможность использовать эти данные для уменьшения неопределенности о чем-либо, данные превращаются в информацию.

Одной из важнейших разновидностей информации является экономическая информация, сопровождающая процессы производства, распределения, обмена и потребления материальных благ и услуг.

**Экономическая информация** – совокупность сведений, отражающих социально-экономические процессы и служащих для управления этими процессами и коллективами людей в производственной и непроизводственной сфере.

При работе с информацией всегда имеется ее источник и потребитель (получатель).

Пути и процессы, обеспечивающие передачу сообщений от источника информации к ее потребителю, называются **информационными коммуникациями**.

## МЕРЫ ИНФОРМАЦИИ.

Классификация мер.

Для измерения информации вводятся два параметра:

- количество информации **I**
- объем данных **V<sub>д</sub>**.

Эти параметры имеют разные выражения и интерпретацию в зависимости от рассматриваемой формы адекватности. Каждой форме адекватности соответствует своя мера количества информации и объема данных.



Рис. 1.2. Классификация мер информации

### Мера Р.Хартли

Пусть имеется  $N$  состояний системы  $S$  или  $N$  опытов с различными, равновероятными последовательными состояниями системы.

Если каждое состояние закодировать, например, двоичными кодами определенной длины  $d$ . То эту длину необходимо выбрать так, чтобы число всех возможных комбинаций было не меньше чем  $N$ .

Наименьшее число, при котором возможно называют **мерой разнообразия** множества состояний системы и задается формулой Р.Хартли

$$H = K \log_a N$$

где:

**k**- коэффициент пропорциональности масштабирования, в зависимости от выбранной единицы измерения меры;

**a**- основание системы меры.

**Синтаксическая мера информации** – эта мера оперирует с обезличенной информацией, не выражающей смыслового отношения к объекту.

**Объем данных** в сообщении измеряется количеством символов (разрядов) в этом сообщении. В различных системах счисления один разряд имеет различный вес и соответственно меняется единица измерения данных (в двоичной СС – бит, в десятичной - дит).

**Количество информации** на синтаксическом уровне невозможно рассмотреть без рассмотрения понятия неопределенности состояния системы (энтропии системы).

Рассмотрим это понятие.

Пусть до получения информации потребитель имеет некоторые предварительные (априорные) сведения о системе  $\alpha$ . Мерой его неосведомленности о системе является функция  $H(\alpha)$ , которая в то же время служит и мерой неопределенности состояния системы.

После получения некоторого сообщения  $\beta$  получатель приобрел некоторую дополнительную информацию  $I_{\beta}(\alpha)$ , уменьшившую его априорную неосведомленность так, апостериорная (после получения сообщения  $\beta$ ) неопределенность состояния системы стала  $H_{\beta}(\alpha)$ .

Тогда количество информации  $I_{\beta}(\alpha)$  о системе, полученной в сообщении  $\beta$ , определяется как

$$I_{\beta}(\alpha) = H(\alpha) - H_{\beta}(\alpha),$$

т.е. количество информации измеряется изменением (уменьшением) неопределенности состояния системы.

Если конечная неопределенность  $H_{\beta}(\alpha)$  обратиться в нуль, то первоначальное неполное знание заменится полным знанием и количество информации  $I_{\beta}(\alpha) = H(\alpha)$ . Иными словами, **энтропия системы  $H(\alpha)$**  может рассматриваться как мера недостающей информации.

**Энтропия системы  $H(\alpha)$** , имеющая  $N$  возможных состояний согласно формуле **Шеннона** равна

$$H(\alpha) = - \sum_{I=1}^N P_I \log P_I ,$$

Где  $P_I$  – вероятность того, что система находится в  $I$ -ом состоянии.

Для случая, когда все состояния системы равны, т.е. их вероятности равны

$P_I = 1/N$ , ее энтропия определяется соотношением :

$$H(\alpha) = - \sum_{I=1}^N \frac{1}{N} \log \frac{1}{N}$$

Часто информация кодируется числовыми кодами в той или иной СС. Естественно, что одно и то же количество разрядов в разных СС может передать разное число состояний отображаемого объекта, что можно представить в виде соотношения:

$$N = m^n, \text{ где:}$$

- N- число возможных отображаемых состояний,
- m - основание СС,
- n - число разрядов(символов) в сообщении.

Наиболее часто используются двоичные и десятичные логарифмы (единицами измерения будут бит и дит).

**Коэффициент (степень) информативности** (лаконичность) сообщения определяется соотношением количества информации к объему данных:

$$Y = I / V_d, \text{ причем } 0 < Y < 1.$$

С увеличением V уменьшаются работы по преобразованию информации (данных) в системе. Поэтому стремятся к повышению информативности, для чего разрабатываются специальные методы оптимального кодирования информации.

### Семантическая мера информации.

**Тезаурусная мера** используется для измерения смыслового содержания информации, т.е. количества ее на семантическом уровне. Она связывает семантические свойства информации со способностью пользователя принимать поступившее сообщение.

**Тезаурус**- это совокупность сведений, которым располагает пользователь или система.

В зависимости от соотношений между смысловым содержанием информации S и тезаурусом пользователя S<sub>p</sub> изменяется количество семантической информации I<sub>c</sub>, воспринимаемой пользователем и включаемой им в дальнейшем в свой тезаурус.

Характер такой зависимости показан на рис. Рассмотрим два предельных случая, когда количество семантической информации I<sub>c</sub> равно 0:

- При  $S_p=0$  пользователь не воспринимает, не понимает поступающую информацию.
- При  $S_p \rightarrow \infty$  пользователь все знает, и поступающая информация ему не нужна.

Максимальное количество семантической информации  $I_c$  потребитель приобретает при согласовании ее смыслового содержания  $S$  со своим тезаурусом  $S_p$  ( $S_p = S_p \text{ opt}$ ), когда поступающая информация понятна пользователю и несет ему ранее не известные (отсутствующие в его тезаурусе) сведения.

Следовательно, количество семантической информации в сообщении, количество новых знаний, получаемых пользователем, является величиной относительной. Одно и то же сообщение может иметь смысловое содержание для компетентного пользователя и быть бессмысленным (семантический шум) для пользователя некомпетентного.

При оценке семантического (содержательного) аспекта информации необходимо стремиться к согласованию величин  $S$  и  $S_p$ .

Относительной мерой количества семантической информации может служить коэффициент содержательности  $C$ , который определяется как отношение количества семантической информации к ее объему:

$$C = \frac{I_c}{V_d}$$

### Прагматическая мера информации.

Эта мера определяет полезность информации (ценность) для достижения пользователем поставленной цели. Эта мера также величина относительная, обусловленная особенностями использования этой информации в той или иной системе. Ценность информации целесообразно измерять в тех же самых единицах (или близких к ним), в которых измеряется целевая функция.

**Пример.** В экономической системе прагматические свойства (ценность) информации можно определить приростом экономического эффекта функционирования, достигнутым благодаря использованию этой информации для управления системой:

$$In\beta(\gamma) = \Pi(\gamma/\beta) - \Pi(\gamma),$$

$In\beta(\gamma)$  – ценность информационного сообщения  $\beta$  для системы управления  $\gamma$ ,

$P(\gamma)$ - априорный ожидаемый экономический эффект функционирования системы управления  $\gamma$ ,

$P(\gamma/\beta)$  – ожидаемый эффект функционирования системы  $\gamma$  при условии, что для управления будет использована информация, содержащаяся в сообщении  $\beta$

Для сопоставления введенные меры информации представим в таблице.

Таблица 1 Меры информации

Мера информации	Единицы измерения	Примеры (для компьютерной области)
Синтаксическая: Шенноновский подход	Степень уменьшения неопределенности	Вероятность события
Компьютерный подход	Единицы представления информации	Бит, Байт Кбайт и т.д.
Семантическая	Тезаурус	ППП, Персональный компьютер, компьютерные сети и т.д.
	Экономические показатели	Рентабельность, производительность компьютера, скорость передачи данных и т.д.
Прагматическая	Ценность использования	Емкость памяти, производительность компьютера, скорость передачи данных и т.д.
		Время обработки информации и принятия решений

В каком виде существует информация?

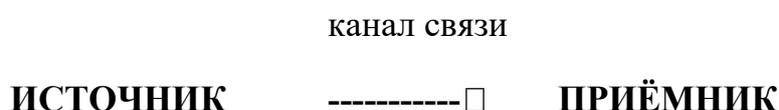
Информация может существовать в виде:

- 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.

## 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**.



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**.

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.

## Что можно делать с информацией?

Информацию можно:

- создавать;
- передавать;
- воспринимать;
- использовать;
- формализовать;
- распространять;
- преобразовывать;
- комбинировать;
- собирать;
- хранить;
- искать;
- измерять;

- запоминать;
- обрабатывать;
- разрушать;
- принимать;
- делить на части;
- и др.
- копировать;
- упрощать;
- 

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

### Какими свойствами обладает информация?

Свойства информации:

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

### КАЧЕСТВО ИНФОРМАЦИИ.

Возможность и эффективность использования информации обуславливаются следующими основными показателями качества:

- **Репрезентативность** информации связана с правильностью ее отбора и формирования в целях адекватного отражения свойств объекта. Важнейшее значение здесь имеют:
  - ◆ Правильность концепции, на базе которой сформулировано исходное понятие.
  - ◆ Обоснованность отбора существенных признаков и связей отображаемого явления.

Нарушение репрезентативности информации приводит нередко к существенным ее погрешностям.

- **Содержательность** информации отражает семантическую емкость, равную отношению количества семантической информации в сообщении к объему обрабатываемых данных, т.е.

$$C = I_c / V_d.$$

С увеличением содержательности информации растет семантическая пропускная способность информационной системы, т.к. для получения одних и тех же сведений требуется преобразовать меньший объем данных.

Наряду с коэффициентом содержательности -С, отражающим **семантический** аспект, можно использовать и коэффициент информативности, характеризующийся отношением количества синтаксической информации ( по Шеннону) к объему данных

$$Y = I / V_d.$$

- **Достаточность (полнота)** информации означает, что она содержит минимальный, но достаточный для принятия правильного решения состав (набор показателей). Понятие полноты информации связано с ее смысловым содержанием (семантикой) и прагматикой. Как неполная т.е. недостаточная для принятия правильного решения, так и избыточная информация снижает эффективность принимаемых пользователем решений.
- **Доступность** информации восприятию пользователя обеспечивается выполнением соответствующих процедур ее получения и преобразования. Например, в информационной системе информация преобразовывается к доступной и удобной для восприятия пользователя форме. Это достигается, в частности, и путем согласования ее семантической формы с тезаурусом пользователя.
- **Актуальность** информации определяется степенью сохранения ценности информации для управления в момент ее использования и зависит от динамики изменения ее характеристик и от интервала времени, прошедшего с момента возникновения данной информации.
- **Своевременность** информации означает ее поступление не позже заранее назначенного момента времени, согласованного с временем поставленной задачи.
- **Точность** информации определяется степенью близости получаемой информации к реальному состоянию объекта, процесса, явления и т.п. Для информации, отображаемой цифровым кодом, известны четыре классификационных понятия точности:
  - ◆ **Формальная точность**, измеряемая значением единицы младшего разряда числа.
  - ◆ **Реальная точность**, определяемая значением единицы последнего разряда числа, верность которого гарантируется.
  - ◆ **Максимальная точность**, которую можно получить в конкретных условиях функционирования системы.
  - ◆ **Необходимая точность**, определяемая функциональным назначением показателя.
- **Достоверность** информации определяется ее свойством отражать реально существующие объекты с необходимой точностью. Измеряется достоверность информации доверительной вероятностью необходимой точности, т.е. вероятностью того, что отображаемое информацией значение параметра отличается от истинного значения этого параметра в пределах необходимой точности.
  - **Устойчивость** информации отражает ее способность реагировать на изменение исходных данных без нарушения необходимой точности. Устойчивость информации, как и репрезентативность, обусловлена выбранной методикой ее отбора и формирования.

Такие параметры качества информации, как **репрезентативность, содержательность, достаточность, доступность, устойчивость**, целиком

определяются на методическом уровне разработки информационных систем, Параметры **актуальности, своевременности, точности и достоверности** обуславливаются в большей степени также на методическом уровне, однако на их величину существенно влияет и характер функционирования системы, в первую очередь ее надежность. При этом параметры актуальности и точности жестко связаны с параметрами своевременности и достоверности.

### Формы представления информации

Различают две формы представления информации — **непрерывную (аналоговую)** и **прерывистую (цифровую, дискретную)**. Непрерывная форма характеризует процесс, который не имеет перерывов и может изменяться в любой момент времени и теоретически на любую величину (например, речь человека, музыкальное произведение). Цифровой сигнал может изменяться лишь в определенные моменты времени и принимать лишь заранее обусловленные значения (например, только значения напряжений 0 и 3,5 В). Моменты возможного изменения уровня цифрового сигнала задает тактовый генератор конкретного цифрового устройства.

Для преобразования аналогового сигнала в цифровой сигнал требуется провести дискретизацию во времени и квантование по уровню.

Дискретизация — замена непрерывного (аналогового) сигнала последовательностью отдельных во времени отсчетов этого сигнала. Наиболее распространена равномерная дискретизация, в основе которой лежит теорема Котельникова.

На рисунке схематично показан процесс преобразования аналогового сигнала в цифровой сигнал. Цифровой сигнал в данном случае может принимать лишь пять различных уровней. Естественно, что качество такого преобразования невысокое. Из рисунка видно, что изменение цифрового сигнала возможно лишь в некоторые моменты времени (в данном случае этих моментов — одиннадцать).

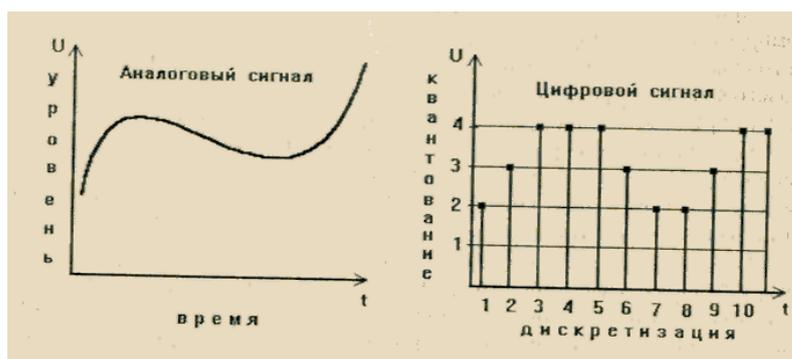


Рис. 3

После такого преобразования непрерывный сигнал представляют последовательностью чисел. Показанный на рисунке непрерывный сигнал заменяется числами: 2-3-4-4-4-3-2-2-3-4-4.

Перечисленные десятичные числа на выходе устройства трансформации непрерывного сигнала в цифровой сигнал кодируют с помощью двоичной системы счисления (единиц и нулей). Результаты данного преобразования можно представить таблицей на рис.4:

Время	Десятичные числа	Двоичные числа
$t_1$	2	0010
$t_2$	3	0011
$t_3$	4	0100
$t_4$	4	0100
$t_5$	4	0100
$t_6$	3	0011
$t_7$	2	0010
$t_8$	2	0010
$t_9$	3	0011
$t_{10}$	4	0100
$t_{11}$	4	0100

Рис.4

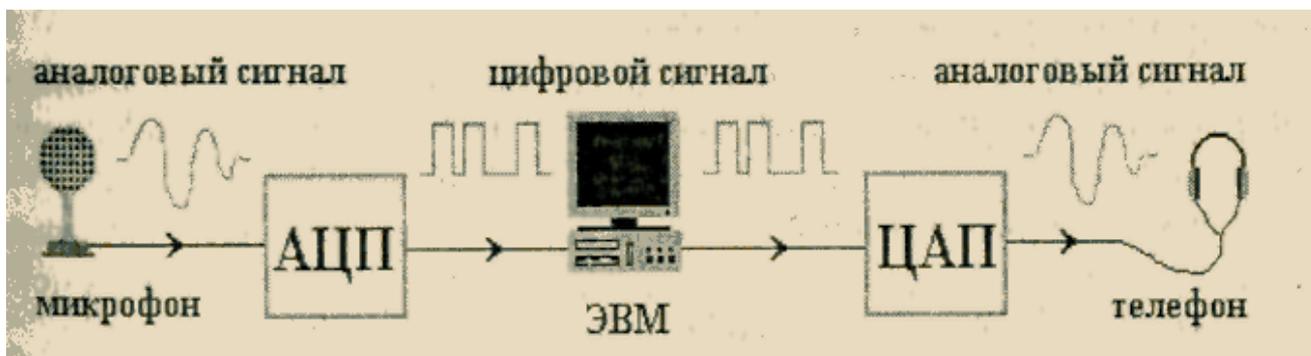
В данном случае цифровые сигналы представлены четырьмя разрядами. Очевидно, что чем больше разрядов у двоичных чисел (а значит тем больше число уровней квантования) и чем чаще во времени осуществляются отсчеты (выборки), тем точнее будет преобразован непрерывный сигнал в цифровой.

Первое представление об аналоговом и цифровом способах хранения и распространения информации можно получить, рассматривая два способа записи звуковых сигналов: аналоговую и цифровую аудиозаписи.

При аналоговой аудиозаписи непрерывный электрический сигнал, формируемый источником звука на выходе микрофона, с помощью магнитной головки наносится на движущуюся магнитную ленту. Недостатком аналогового способа обработки информации является то, что копия бывает всегда хуже оригинала.

При цифровой аудиозаписи используется процесс выборки, заключающийся в периодическом измерении уровня (громкости) аналогового звукового сигнала (например, поступающего с выхода микрофона) и превращении полученного значения в последовательность двоичных чисел. Для преобразования аналогового сигнала в цифровой используется

специальный конвертор, называемый аналого-цифровой преобразователь (АЦП). Сигнал на выходе АЦП представляет собой последовательность двоичных чисел, которая может быть записана на лазерный диск или обработана компьютером. Обратная конверсия цифрового сигнала в непрерывный сигнал осуществляется с помощью цифроаналогового преобразователя (ЦАП).



Качество аналого-цифрового преобразования характеризует параметр, называемый разрешением. Разрешение — это количество уровней квантования, используемых для замены непрерывного аналогового сигнала цифровым сигналом. Восемьразрядная выборка позволяет получить только 256 различных уровней квантования цифрового сигнала, а шестнадцатиразрядная выборка — 65536 уровней.

Еще один показатель качества трансформации непрерывного сигнала в цифровой сигнал — это частота дискретизации — количество преобразований аналог-цифра (выборок), производимое устройством в одну секунду.

Этот показатель измеряют килогерцами (килогерц — тысяча выборок в секунду). Типичное значение частоты дискретизации современных аудиодисков — 44,1 кГц.

На рисунке слева показан; структура лазерного аудиодиска. Все звуки преобразованы в последовательность единичек и нулей, которые на диске выглядят как выступы и впадины. При этом копию можно получить такого же качества, как и оригинал.

Имеется тенденция перехода к единому цифровому представлению всех видов информации. Глобальная сеть Интернет претендует на то, чтобы объединить все средства вещания и коммуникации- компьютерные, телефонные, радио- и видео сети, связав их в единое «киберпространство».

С позиции каждого отдельного человека количество информации, содержащееся в каком-либо сообщении — субъективная величина.

Объективная количественная мера информации может быть введена на основе вероятностной трактовки информационного обмена.

Этот способ измерения количества информации впервые предложил в 1948 г. К. Шеннон. По К. Шеннону, информация — это сведения, уменьшающие неопределенность (энтропию), существовавшую до их получения.

Наименьшей единицей информации является бит (от англ. binary digit—двоичный разряд]. Сообщение о том, что произошло одно из двух возможных событий, дает получателю один бит информации.

Один бит информации получает человек, когда он узнает: опаздывает с прибытием нужный ему поезд или нет, был ночью мороз или нет, присутствует на лекции студент Иванов или нет и т.д.

Более крупная единица информации — байт равна 8 бит. Проверка присутствия или отсутствия на лекции 24 студентов дает лектору три байта информации. Еще более крупная единица информации

1 Кбайт равна 1024 байтам.  $2^{10}$ -ой степени

1 Мбайт равен 1024 Кбайт.  $2^{20}$ -ой степени

1 Гбайт равен 1024 Мбайт  $2^{30}$ -ой степени

1 Тбайт равен 1024 Гбайт  $2^{40}$ -ой степени

Перечисленные единицы измерения информации, произносятся так: Кбайт — килобайт, Мбайт —мегабайт, Гбайт — гигабайт, Тбайт — терабайт.

### Понятие о единицах измерения информации в ЭВМ:

Любое число несет некоторое кол-во информации о соответствующей ему величине.

Кол-во информации, которую несет одноразрядное двоичное число (1 разряд двоичного числа), в информатике называют бит. Еще можно сказать, что бит – минимальная единица измерения информации.

ПРИМЕР: - двоичное число 01 несет 2 бита информации

- 10101 несет пять бит информации.

- 8-разрядное двоичное число из 8 знаков несет 8 бит информации, такое кол-во информации называется байт – еще одна единица кол-во информации.

Любое запоминающее устройство ЭВМ состоит из ячеек, в каждой из которых можно записать 8-ми разрядное двоичное число, т.е. равно – байту информации. Такая ячейка также называется байт. В нее можно поместить натуральное число не более чем 256 ( $2^8$  в 8 степени).

Из байтов, как из кирпичиков, можно складывать более крупные ячейки, но в каждую из них записывается только одно число. Так, можно создать ячейки из 2-х, из 4-х байт – в них можно поместить и большие числа – натуральное число до  $2^{16}$  в 16-ой и до  $2^{32}$  в 32-ой степени соответственно. В ЭВМ используются ячейки, содержащие тысячи байт. При размещении текста в ячейки каждый его символ (буква, цифра, точка, пробел и т.д.) занимает 1 байт.

Таким образом, **байт** – основная единица объема памяти ЭВМ. В современных ЭВМ используются более крупные единицы измерения объема памяти:

1 килобит= $2^7$ -ой степени = 128 байт

1 килобайт=1024 байта ( $2^{10}$ -ой степени)

1 мегабайт=1024 килобайта

1 гигабайт=1024 мегабайт

Сколько значений кодируют единицы измерения информации:

Бит – 2 значения (0 или 1)  $2^1$ -ой степени

Байт – 256 значений  $2^8$ -ой степени

## 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.

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

### 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.

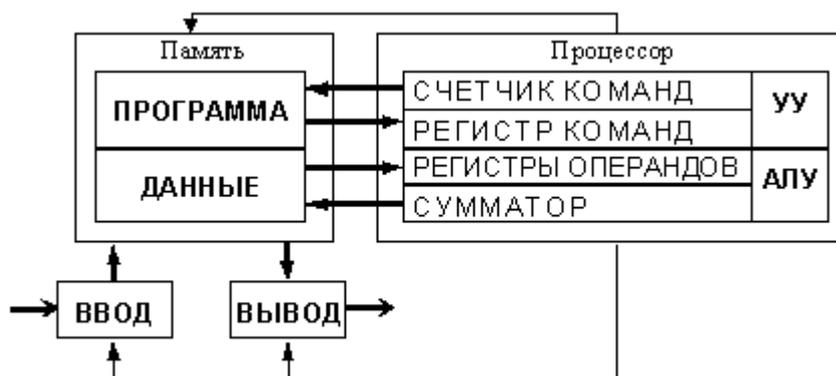


Рис. 2.1. Общая схема компьютера

### 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 определённым 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**.

На каких принципах построены компьютеры?

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

**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**.

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;
- переменнаадресные.

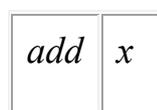
Команды хранятся в ячейках памяти в двоичном коде.

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

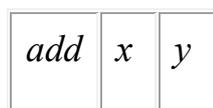
- сам операнд (число или символ);
- адрес операнда (номер байта, начиная с которого расположен операнд);
- адрес адреса операнда (номер байта, начиная с которого расположен адрес операнда), и др.

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

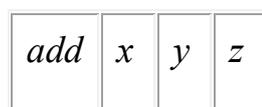
- **одноадресная команда *add x*** (содержимое ячейки *x* сложить с содержимым сумматора, а результат оставить в сумматоре)



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

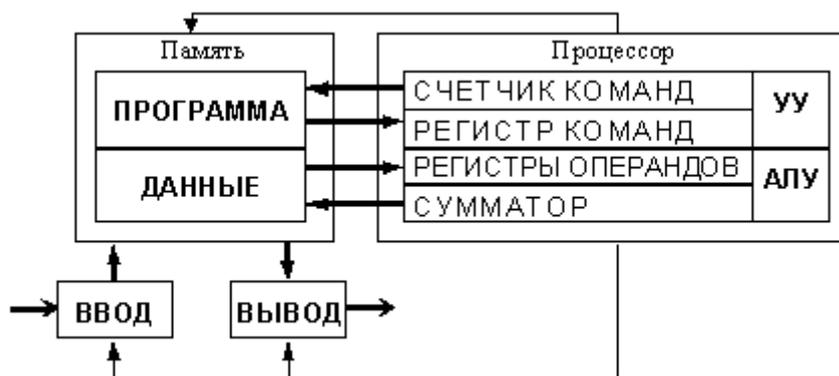


- **трехадресная команда *add x, y, z*** (содержимое ячейки *x* сложить с содержимым ячейки *y*, сумму поместить в ячейку *z*)



Как выполняется команда?

Выполнение команды можно проследить по схеме:



Общая схема компьютера

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".

### 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.

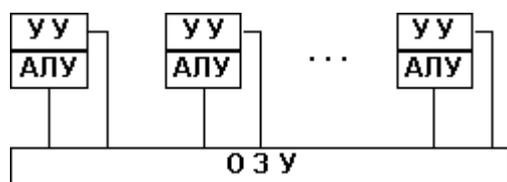


Рис. 2.3. Архитектура многопроцессорного компьютера

**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.

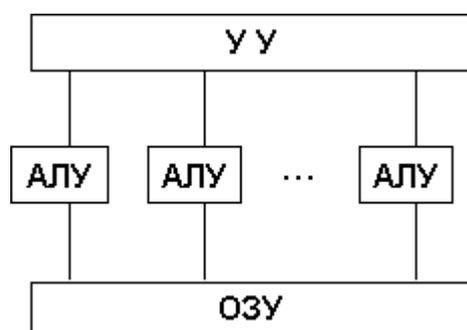


Рис. 2.4. Архитектура с параллельным процессором

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

### 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 **микропроцессоров**. Физически микропроцессор 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 видеоматериалов, распознавание речи, обработка трехмерной графики, игры.

Рис.2.5

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

### 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
<b>ПОЛУСЛОВО</b>		<b>ПОЛУСЛОВО</b>		<b>ПОЛУСЛОВО</b>		<b>ПОЛУСЛОВО</b>	

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**.

### Какие устройства образуют внутреннюю память?

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

#### 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**.

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**.

#### Кэш-память

*Кэш* (англ. cache), или сверхоперативная память — очень быстрое ЗУ небольшого объёма, 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.

#### 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.

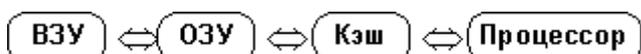
Содержимое CMOS изменяется специальной программой **Setup**, находящейся в BIOS (англ. Set-up — устанавливать, читается "сетап").

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.

## 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.

### 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.

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.

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  $360 \text{ мин}^{-1}$ . Накопитель связан с процессором через **контроллер гибких дисков**.

### Накопители на жестких магнитных дисках

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

*Накопитель на жёстких 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.

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 подъёме 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**.

### 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.

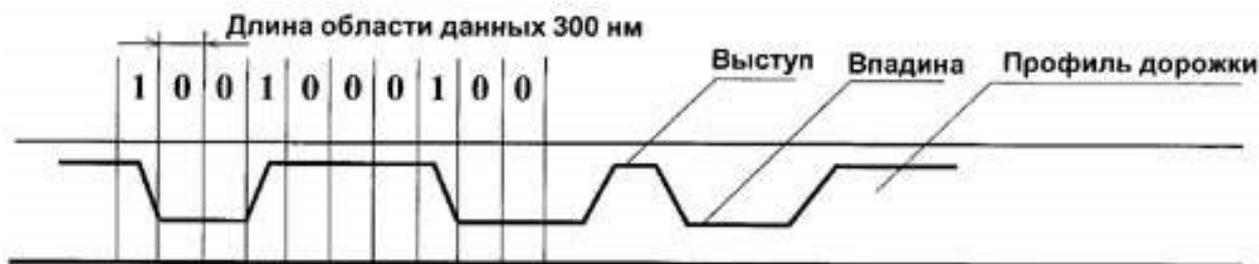
Информация на диске представляется в виде последовательности **впадин** (углублений в диске) и **выступов** (их уровень соответствует поверхности диска), расположенных на спиральной дорожке, выходящей из области вблизи оси диска. На каждом дюйме (2,54 см) по радиусу диска размещается 16 тысяч витков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

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.



Профиль дорожки CD-ROM

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.

### 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.

**Накопитель на магнито-оптических компакт-дисках CD-МО** (Compact Disk — Magneto Optical) (рис. 2.10). Диски CD-МО 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.

### 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.

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.

### 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.

[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:**

**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 видеомаягнитофона, 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.

## 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 (курсор — 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:

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.

### 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.).

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.

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**.

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.

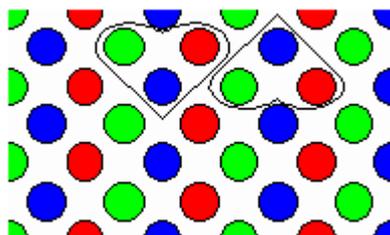


Рис. 2.16. Пиксельные триады

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.

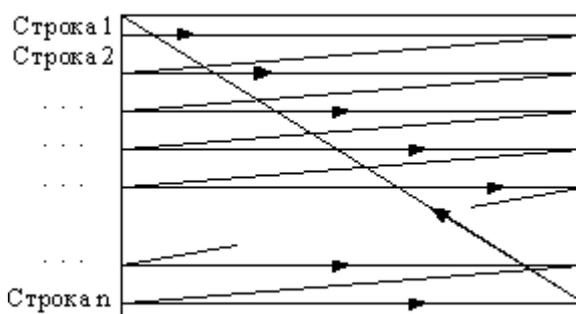


Рис. 2.17. Ход электронного пучка по экрану

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 пикселы 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**.

#### 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 введённые 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).

**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.

#### 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.

#### 2Что 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.

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.

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.

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**.

### 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.

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.**

### 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 коробка 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.

## 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.

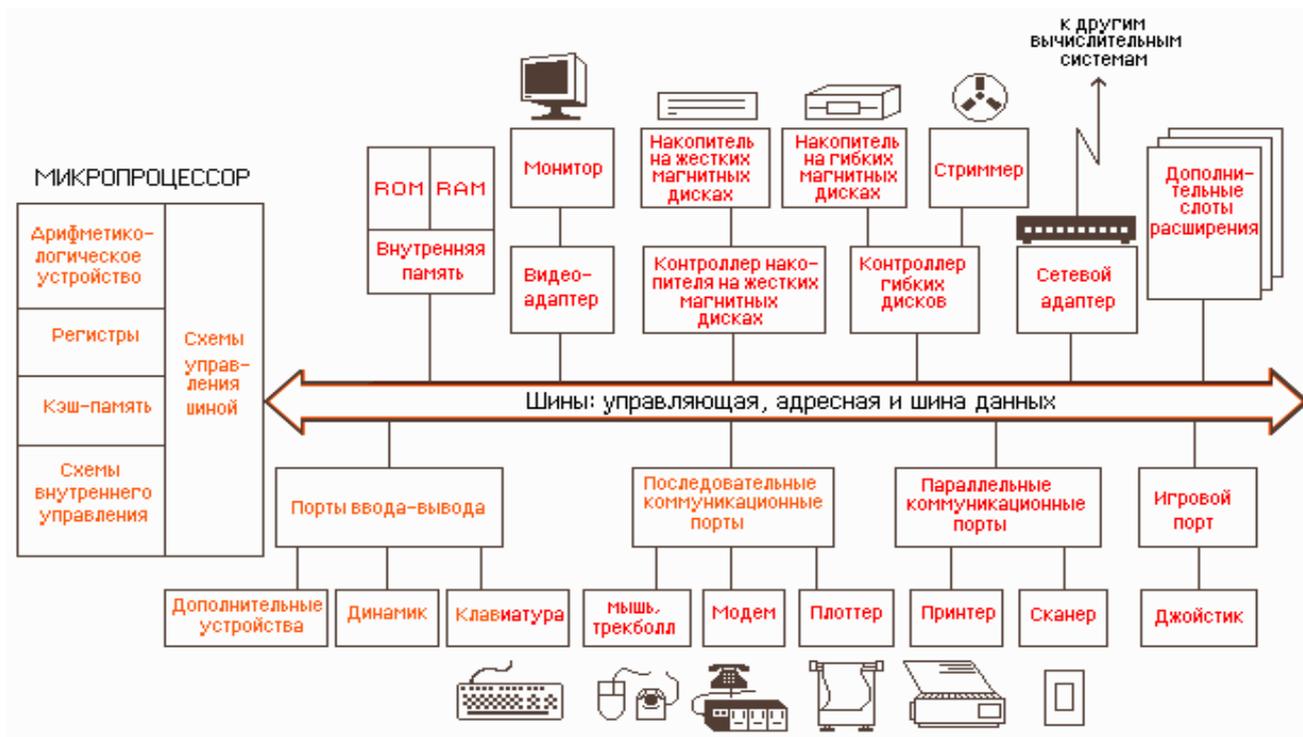


Рис. 2.26. Общая структура персонального компьютера с подсоединенными периферийными устройствами

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).

### Какие основные блоки входят в состав компьютера?

Современный персональный компьютер состоит из нескольких основных конструктивных компонент:

- 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.

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.

## Информация и ее свойства

### Основные определения

Информатика - (от французского information - информация и automatique - автоматика) - это область научно-технической деятельности, занимающаяся исследованием процессов получения, передачи, обработки, хранения и представления информации, решением проблем создания, внедрения и использования информационной техники и технологии во всех сферах общественной жизни.

Основная задача информатики заключается в определении общих закономерностей, в соответствии с которыми происходит создание научной информации, ее преобразование, передача и использование в различных сферах деятельности человека. Прикладные задачи заключаются в разработке более эффективных методов и средств осуществления информационных процессов, в определении способов оптимальной научной коммуникации с широким применением технических средств.

В структуре информатики как науки выделяют алгоритмическую, программную техническую области. Информатика входит в состав кибернетики, изучающей общую теорию управления и передачи информации. Кибернетика - наука об общих законах получения, хранения, передачи и обработки информации в сложных системах. Под сложными системами понимаются технические, биологические и социальные системы. Кибернетика пригодна для исследования любой системы, которая может записывать, накапливать и обрабатывать информацию, благодаря чему ее можно использовать в целях управления.

Информационная система - это организованная человеком система сбора, хранения, обработки и выдачи информации, необходимой для эффективного функционирования субъектов и объектов управления. Данные системы являются средством удовлетворения потребностей управления в информации, которое заключается в том, чтобы в нужный момент из соответствующих источников получать информацию, которая должна быть предварительно систематизирована и определенным образом обработана.

К компонентам информационной системы относятся:

- информация, необходимая для выполнения одной или нескольких функций управления;
- персонал, обеспечивающий функционирование информационной системы;
- технические средства;
- методы и процедуры сбора и переработки информации.

Информационные системы, как и любые другие системы, помимо структуры характеризуются функциями, которые они выполняют. С технологической точки зрения их функции делятся на подготовительные и основные. Подготовительные заключаются в фиксации, сборе данных, кодировании и записи их на машинные носители, вводе в память электронно-вычислительных машин и систематизированном хранении. Основные функции сводятся к поиску или содержательной обработке информации, документальному оформлению и размножению результатов поиска и обработки, передаче выходной информации потребителям.

Информационные системы делятся на три класса:

- не производящие качественного изменения информации (учетные, следящие, прогнозирующие, справочные системы);
- анализирующие информацию (аналитические, советующие, прогнозирующие, диагностические системы);

- вырабатывающие решения (управляющие, планирующие системы).

Информационные системы включают в себя функциональные компоненты, компоненты системы обработки данных, организационные компоненты. Функциональные компоненты представляют собой подсистемы обработки информационных ориентиров по определенному функциональному признаку. Например, бухгалтерский учет и отчетность; технико-экономическое планирование; бизнес-план и т.д. Компоненты системы обработки данных предназначены для сбора и переноса информации на машинные носители, ввода информации в ЭВМ и контроля ввода, создания и ведения внутримашинной информационной базы, обработки информации на ЭВМ (накопление, сортировка, корректировка, выборка, арифметическая и логическая обработка), вывода информации; управления вычислительными процессами в вычислительных сетях. Организационные компоненты информационной системы представляют собой совокупность методов и средств для совершенствования организации структурных объектов, штатного расписания, должностных инструкций персонала.

Информационные технологии – это целенаправленный процесс преобразования информации, использующий совокупность средств и методов сбора, обработки, хранения и передачи информации. Как и многие другие технологии, информационная технология должна отвечать следующим требованиям:

обеспечивать высокую степень деления всего процесса обработки информации на составляющие компоненты;

включать весь набор инструментов, необходимых для достижения поставленной цели;

отдельные компоненты должны быть стандартизированы и унифицированы.

Информатизация – внедрение информационных технологий во все сферы человеческой деятельности

Инфосфера – совокупное информационное пространство.

## Классификация информации

1. Информация подразделяется по форме представления на 2 вида:

- дискретная форма представления информации - это последовательность символов, характеризующая прерывистую, изменяющуюся величину (количество дорожно-транспортных происшествий, количество тяжких преступлений и т.п.);

- аналоговая или непрерывная форма представления информации - это величина, характеризующая процесс, не имеющий перерывов или промежутков (температура тела человека, скорость автомобиля на определенном участке пути и т.п.).

2. По области возникновения выделяют информацию:

- элементарную (механическую), которая отражает процессы, явления неодушевленной природы;

- биологическую, которая отражает процессы животного и растительного мира;

- социальную, которая отражает процессы человеческого общества.

3. По способу передачи и восприятия различают следующие виды информации:

- визуальную, передаваемую видимыми образами и символами;

- аудиальную, передаваемую звуками;

- тактильную, передаваемую ощущениями;

- органолептическую, передаваемую запахами и вкусами;

- машинную, выдаваемую и воспринимаемую средствами вычислительной техники.

4. Информацию, создаваемую и используемую человеком, по общественному назначению можно разбить на три вида:

- личную, предназначенную для конкретного человека;

- массовую, предназначенную для любого желающего ее пользоваться (общественно-политическая, научно-популярная и т.д.) ;

- специальную, предназначенную для использования узким кругом лиц, занимающихся решением сложных специальных задач в области науки, техники, экономики.

5. По способам кодирования выделяют следующие типы информации:

- символьную, основанную на использовании символов - букв, цифр, знаков и т. д. Она является наиболее простой, но практически применяется только для передачи несложных сигналов о различных событиях. Примером может служить зеленый свет уличного светофора, который сообщает о возможности начала движения пешеходам или водителям автотранспорта.

- текстовую, основанную на использовании комбинаций символов. Здесь так же, как и в предыдущей форме, используются символы: буквы, цифры, математические знаки. Однако информация заложена не только в этих символах, но и в их сочетании, порядке следования. Так, слова КОТ и ТОК имеют одинаковые буквы, но содержат различную информацию. Благодаря взаимосвязи символов и отображению речи человека текстовая информация чрезвычайно удобна и широко используется в деятельности человека: книги, брошюры, журналы, различного рода документы, аудиозаписи кодируются в текстовой форме.

- графическую, основанную на использовании произвольного сочетания в пространстве графических примитивов. К этой форме относятся фотографии, схемы, чертежи, рисунки, играющие большое значение в деятельности человек.

Свойства информации можно рассматривать в трех аспектах: техническом - это точность, надежность, скорость передачи сигналов и т.д.; семантическом - это передача смысла текста с помощью кодов и прагматическом - это насколько эффективно информация влияет на поведение объекта.

### Количественное измерение информации

Двоичные символы могут кодироваться любым способом: буквами А, Б; словами ДА, НЕТ, двумя устойчивыми состояниями системы и т.д. Однако ради простоты записи были взяты цифры 1 и 0. Обработка информации в ЭВМ основана на обмене электрическими сигналами между различными устройствами машины. В компьютере, хранящем, либо обрабатывающем информацию, рассматриваемые символы 0 и 1 могут также обозначаться по-разному: один из них - наличием в рассматриваемом элементе электрического тока, либо магнитного поля, второй - отсутствием электрического тока, либо магнитного поля.

Таким образом, в ЭВМ реализуются два устойчивых состояния. Эти два устойчивых состояния информационной системы определяют единицу измерения информации, называемую БИТОМ. Количество информации, кодируемое двоичной цифрой - 0 или 1, называется битом. Благодаря введению понятия единицы информации появилась возможность определения размера любой информации числом битов.

Процесс получения двоичной информации об объектах исследования называют кодированием информации. Кодирование информации перечислением всех возможных событий очень трудоемко. Поэтому на практике кодирование осуществляется более простым способом. Он основан

на том, что один разряд последовательности двоичных цифр имеет уже вдвое больше различных значений - 00, 01, 10, 11, чем одноразрядные 0 и 1. Трехразрядная последовательность имеет также вдвое больше значений - 000, 001, 010, 011, 100, 101, 110, 111, чем двухразрядная и т.д. Добавление одного разряда увеличивает число значений вдвое, это позволяет составить следующую таблицу информационной емкости чисел:

Таблица 2. Информационная емкость чисел

1 p	2 p	3 p	4 p	5 p	6 p	7 p	8 p
2	4	8	16	32	64	128	256
9 p	10 p	11 p	12 p	13 p	14 p	15 p	16 p
512	1024	2048	4096	8192	16384	32768	65536

Пользуясь вышеприведенной таблицей легко закодировать любое множество событий. Например, нам нужно закодировать 32 буквы русского алфавита, для этой цели достаточно взять пять разрядов, потому что пятиразрядная последовательность имеет 32 различных значения.

Для измерения больших объемов информации пользоваться битами неудобно. Поэтому применяются кратные биту единицы измерения информации:

$$1 \text{ байт} = 2^3 \text{ бит} = 8 \text{ бит}$$

$$1 \text{ кбайт} = 2^{10} \text{ байт} = 1024 \text{ байт}$$

$$1 \text{ Мбайт} = 2^{10} \text{ кбайт} = 1024 \text{ кбайт}$$

$$1 \text{ Тбайт} = 2^{10} \text{ Мбайт} = 1024 \text{ Мбайт}$$

### Кодирование различных типов информации

С помощью набора битов, можно представить любое число и любой знак. В информационных документах широко используются не только русские, но и латинские буквы, цифры, математические знаки и другие специальные знаки, всего их количество составляет примерно 200-250 символов. Поэтому для кодировки всех указанных символов используется восьмиразрядная последовательность цифр 0 и 1. Таким образом, текстовая информация кодируется с помощью кодовой таблицы.

Кодовая таблица – это внутреннее представление символов в компьютере. Во всем мире в качестве стандарта принята таблица ASCII – Американский стандартный код для обмена информацией. Для хранения двоичного кода одного символа выделен 1 байт = 8 бит.

Следует отметить, что указанный способ кодирования используется тогда, когда к нему не предъявляются дополнительные требования, такие как необходимость указать на возникшую ошибку, исправление ошибки, секретность информации. При специальном кодировании коды получаются длиннее, чем в указанной таблице.

Наиболее просто кодируется числовая информация – она переводится в двоичную систему исчисления.

Для представления графической информации в двоичной форме используется так называемый поточечный способ. На первом этапе вертикальными и горизонтальными линиями делят изображение. Чем больше при этом получилось квадратов, тем точнее будет передана информация о картинке. Как известно из физики, любой цвет может быть представлен в виде суммы различной яркости зеленого, синего, красного цветов. Поэтому информация о каждой клетке должна содержать кодировку значения яркости и количеств зеленого, синего и красного компонентов. Таким образом кодируется растровое изображение – изображение, разбитое на отдельные точки. Объем растрового изображения определяется умножением количества точек на рисунке на информационный объем одной точки, который зависит от количества возможных цветов отображения (для черно-белого изображения информационный объем одной точки равен 1 биту и кодируется двумя цифрами – 0 или 1). Разные цвета и их оттенки получаются за счет наличия или отсутствия трех основных цветов – красного, синего, зеленого и их яркости. Каждая точка на экране кодируется с помощью 4 битов.

Векторное изображение кодируется разбиением рисунка на элементарные отрезки, геометрические фигуры и дуги. Положение этих элементарных объектов определяется координатами точек. Для каждой линии указывается ее тип (сплошная, пунктирная, штрих - пунктирная), толщина и цвет. Информация о векторном изображении кодируется как обычная буквенно-цифровая и обрабатывается специальными программами.

Звуковая информация может быть представлена последовательностью элементарных звуков и пауз между ними. Вывод звуков из компьютера осуществляется синтезатором речи, который считывает из памяти хранящийся код звука. Речь человека имеет большое разнообразие оттенков, поэтому каждое произнесенное слово должно сравниваться с предварительно

занесенным в память компьютера эталоном, и при их совпадении происходит его распознавание и запись.

## ПРИМЕРЫ ТЕСТОВ

### Задание № 2

Сообщение содержит 4096 символов. Объем сообщения при использовании равномерного кода составил 1/512 Мбайт. Мощность алфавита, с помощью которого записано данное сообщение, равна ...

#### Решение

Мощность алфавита - количество символов в алфавите. Переведем информационный объем сообщения в биты.

$$\frac{1}{512} \text{ (Мбайт)} = \frac{1}{512} \cdot 1024 \cdot 1024 \cdot 8 = 16384 \text{ (бит)}.$$

Для кодирования одного символа отводится  Тогда мощность алфавита ( $N$ ) по формуле Р. Хартли равна  $N = 2^i = 2^4 = 16$ .

### Задание № 3

Объем видеопамати, необходимой для хранения битовой карты изображения разрешением 1024 x 768 точек в режиме «true color» ( $16777216 = 2^{24}$  цветов), больше, чем в режиме 256 цветов при разрешении экрана 1280 x 1024 точки, на \_\_\_\_\_ Мбайт.

#### Решение

1. Количество бит на точку (пиксель) режима «true color» равно 

2. Рассчитаем объем видеопамати для хранения битовой карты 1024 x 768 \_\_\_\_\_ точек.

$$= \frac{1024 \cdot 768 \cdot 3}{1024} \text{ Кбайт} = \frac{768 \cdot 3}{1024} \text{ Мбайт} = 2,25 \text{ Мбайт}.$$

2. Аналогично для режима 256 цветов.

$$I_{256} = \log_2 256 = 8 \text{ бит}.$$


3. Объем видеопамати в первом случае больше  
на  $2,25 - 1,25 = 1$  Мбайт.
3. Ответ: 1.

### Задание № 2

Скорость передачи данных через ADSL-соединение равна 256000 бит/сек. Передача файла через это соединение по времени заняла 2 мин. Определите размер файла в килобайтах.

#### Решение

Ответ требуется получить в килобайтах, поэтому переведем скорость модема в килобайты/сек (делим на 8, чтобы получить байты, и еще на 1024, чтобы получить килобайты).

$$\frac{256000}{8 \cdot 1024} \text{ (бит/сек)} = \frac{125}{4} \text{ (Кбайт/сек)}.$$

Передача длилась 2 мин, или 120 сек. Находим количество килобайтов:



Правильный ответ: 3750 Кбайт.

### Задание № 3

Количество цветов, воспроизводимых на экране сотового телефона, равно 1024, разрешение экрана 128\*64. Минимальный объем видеопамати равен \_\_\_\_\_ Кбайт.

Графическая информация на экране дисплея представляется в виде изображения, которое формируется из точек (пикселей). В современных компьютерах и сотовых телефонах разрешающая способность (количество точек на экране дисплея), а также количество цветов зависят от видеоадаптера.

Цветные изображения могут иметь различные режимы: 16 цветов, 256 цветов, 1024 цвета, 65536 цветов (high color), 16777216 цветов (true color). Очевидно, что количество бит на точку (пиксель), например, режима

«high color», равно:  $I = \log_2 65536 = \log_2 2^{16} = 16$  (бит).

Рассчитаем количество информации, приходящейся на одну точку (пиксель) экрана сотового телефона нашей

задачи:  $I = \log_2 1024 = \log_2 2^{10} = 10$  (бит).

По условию задачи экран имеет размер 128\*64, т.е. всего на экране 128\*64 =

8192

(точек).

Рассчитаем

необходимый

объем

$$V = 10(\text{бит}) \cdot 8192 = 81920$$

видеопамяти:

бит.

Знаем: 1 Кбайт = 1024 байт, 1 байт = 8 бит.

Тогда:

## Задание № 2

В конкурсе участвовали 20 студентов, 8 школьников и 4 учащихся колледжа. Количество информации в сообщении о том, что победил школьник, считая, что победа любого из участников равновероятна, составит \_\_\_\_ бит(-а).

Рассчитаем вероятность того, что в конкурсе победил школьник.

$$p = \frac{m}{n},$$

Для этого воспользуемся формулой классической вероятности:

где  $m$  - число элементарных исходов, благоприятных событию (победил школьник), т.е. число школьников, участвовавших в конкурсе;  $n$  - общее число всех элементарных равновозможных исходов опыта, т.е. общее число всех участников конкурса.

$$p = \frac{8}{20 + 8 + 4} = \frac{8}{32} = \frac{1}{4}.$$

Воспользуемся формулой Хартли для вычисления искомого количества

$$I = \log_2 \left( \frac{1}{p} \right).$$

информации:

## Основные понятия систем счисления

Система счисления - это совокупность правил и приемов записи чисел с помощью набора цифровых знаков. Количество цифр, необходимых для записи числа в системе, называют основанием системы счисления. Основание системы записывается в справа числа в нижнем индексе:  $5_{10}$ ;  $1110110_2$ ;  $AF178_{16}$  и т. д.

Различают два типа систем счисления:

- позиционные, когда значение каждой цифры числа определяется ее позицией в записи числа;
- непозиционные, когда значение цифры в числе не зависит от ее места в записи числа.

Примером непозиционной системы счисления является римская: числа IX, IV, XV и т.д. Примером позиционной системы счисления является десятичная система, используемая повседневно.

Любое целое число в позиционной системе можно записать в форме многочлена:

$$X_S = \{A_n A_{n-1} A_{n-2} \dots A_2 A_1\}_S = A_n \cdot S^{n-1} + A_{n-1} \cdot S^{n-2} + A_{n-2} \cdot S^{n-3} + \dots + A_2 \cdot S^1 + A_1 \cdot S^0$$

где S - основание системы счисления;

$A_n$  - цифры числа, записанного в данной системе счисления;

n - количество разрядов числа.

Пример. Число  $6293_{10}$  запишется в форме многочлена следующим образом:

$$6293_{10} = 6 \cdot 10^3 + 2 \cdot 10^2 + 9 \cdot 10^1 + 3 \cdot 10^0$$

## Виды систем счисления

Римская система счисления является непозиционной системой. В ней для записи чисел используются буквы латинского алфавита. При этом буква I всегда означает единицу, буква V - пять, X - десять, L - пятьдесят, C - сто, D - пятьсот, M - тысячу и т.д.

Например, число 264 записывается в виде CCLXIV.

При записи чисел в римской системе счисления значением числа является алгебраическая сумма цифр, в него входящих. При этом цифры в записи числа следуют, как правило, в порядке убывания их значений, и не разрешается записывать рядом более трех одинаковых цифр. В том случае, когда за цифрой с большим значением следует цифра с меньшим, ее вклад в значение числа в целом является отрицательным. Типичные примеры, иллюстрирующие общие правила записи чисел в римской системе счисления, приведены в таблице.

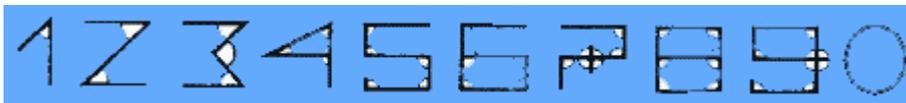
Таблица 3. Запись чисел в римской системе счисления

1	2	3	4	5
I	II	III	IV	V
6	7	8	9	10
VI	VII	VIII	IX	X
11	13	18	19	22
XI	XIII	XVIII	XIX	XXII
34	39	40	60	99
XXXIV	XXXIX	XL	LX	XCIX
200	438	649	999	1207
CC	CDXXXVIII	DCXLIX	CMXCIX	MCCVII
2045	3555	3678	3900	3999
MMXLV	MMMDLV	MMMDCCLXXVIII	MMMCM	MMMCMXCIX

Недостатком римской системы является отсутствие формальных правил записи чисел и, соответственно, арифметических действий с многозначными числами. По причине неудобства и большой сложности в настоящее время римская система счисления используется там, где это действительно удобно: в литературе (нумерация глав), в оформлении документов (серия паспорта, ценных бумаг и др.), в декоративных целях на циферблате часов и в ряде других случаев.

Десятичная система счисления – в настоящее время наиболее известная и используемая. Изобретение десятичной системы счисления относится к главным достижениям человеческой мысли. Без нее вряд ли могла существовать, а тем более возникнуть современная техника. Причина, по которой десятичная система счисления стала общепринятой, вовсе не математическая. Люди привыкли считать в десятичной системе счисления, потому что у них по 10 пальцев на руках.

Древнее изображение десятичных цифр (рис. 1) не случайно: каждая цифра обозначает число по количеству углов в ней. Например, 0 - углов нет, 1 - один угол, 2 - два угла и т.д. Написание десятичных цифр претерпело существенные изменения. Форма, которой мы пользуемся, установилась в XVI веке.



Десятичная система впервые появилась в Индии примерно в VI веке новой эры. Индийская нумерация использовала девять числовых символов и нуль для обозначения пустой позиции. В ранних индийских рукописях, дошедших до нас, числа записывались в обратном порядке - наиболее значимая цифра ставилась справа. Но вскоре стало правилом располагать такую цифру с левой стороны. Особое значение придавалось нулевому символу, который вводился для позиционной системы обозначений. Индийская нумерация, включая нуль, дошла и до нашего времени. В Европе индусские приёмы десятичной арифметики получили распространение в начале XIII в. благодаря работам итальянского математика Леонардо Пизанского (Фибоначчи). Европейцы заимствовали индийскую систему счисления у арабов, назвав ее арабской. Это исторически неправильное название удерживается и поныне.

Десятичная система использует десять цифр – 0, 1, 2, 3, 4, 5, 6, 7, 8 и 9, а также символы “+” и “-” для обозначения знака числа и запятую или точку для разделения целой и дробной частей числа.

В вычислительных машинах используется двоичная система счисления, её основание - число 2. Для записи чисел в этой системе используют только две цифры - 0 и 1. Вопреки распространенному заблуждению, двоичная система счисления была придумана не инженерами-конструкторами ЭВМ, а математиками и философами задолго до появления компьютеров, еще в XVII - XIX веках. Первое опубликованное обсуждение двоичной системы счисления принадлежит испанскому священнику Хуану Карамюэлю Лобковицу (1670 г.). Всеобщее внимание к этой системе привлекла статья немецкого математика Готфрида Вильгельма Лейбница, опубликованная в 1703 г. В ней пояснялись двоичные операции сложения, вычитания, умножения и деления. Лейбниц не рекомендовал использовать эту систему для практических вычислений, но подчёркивал её важность для теоретических исследований. Со временем двоичная система счисления становится хорошо известной и получает развитие.

Выбор двоичной системы для применения в вычислительной технике объясняется тем, что электронные элементы - триггеры, из которых состоят микросхемы ЭВМ, могут находиться только в двух рабочих состояниях.

С помощью двоичной системы кодирования можно зафиксировать любые данные и знания. Это легко понять, если вспомнить принцип кодирования и передачи информации с помощью азбуки Морзе. Телеграфист, используя только два символа этой азбуки - точки и тире, может передать практически любой текст.

Двоичная система удобна для компьютера, но неудобна для человека: числа получаются длинными и их трудно записывать и запоминать. Конечно, можно перевести число в десятичную систему и записывать в таком виде, а потом, когда понадобится перевести обратно, но все эти переводы трудоёмки. Поэтому применяются системы счисления, родственные двоичной - восьмеричная и шестнадцатеричная. Для записи чисел в этих системах требуется соответственно 8 и 16 цифр. В 16-теричной первые 10 цифр общие, а дальше используют заглавные латинские буквы. Шестнадцатеричная цифра А соответствует десятичному числу 10, шестнадцатеричная В – десятичному числу 11 и т. д. Использование этих систем объясняется тем, что переход к записи числа в любой из этих систем от его двоичной записи очень прост. Ниже приведена таблица соответствия чисел, записанных в разных системах.

Таблица 4. Соответствие чисел, записанных в различных системах счисления

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
1	001	1	1
2	010	2	2
3	011	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	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

## Правила перевода чисел из одной системы счисления в другую

Перевод чисел из одной системы счисления в другую составляет важную часть машинной арифметики. Рассмотрим основные правила перевода.

1. Для перевода двоичного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 2, и вычислить по правилам десятичной арифметики:

$$X_2 = A_n \cdot 2^{n-1} + A_{n-1} \cdot 2^{n-2} + A_{n-2} \cdot 2^{n-3} + \dots + A_2 \cdot 2^1 + A_1 \cdot 2^0$$

При переводе удобно пользоваться таблицей степеней двойки:

Таблица 4. Степени числа 2

n (степен ь)	0	1	2	3	4	5	6	7	8	9	10
$2^n$	1	2	4	8	16	32	64	128	256	512	1024

Пример . Число  $11101000_2$  перевести в десятичную систему счисления.

$$11101000_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 232_{10}$$

2. Для перевода восьмеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 8, и вычислить по правилам десятичной арифметики:

$$X_8 = A_n \cdot 8^{n-1} + A_{n-1} \cdot 8^{n-2} + A_{n-2} \cdot 8^{n-3} + \dots + A_2 \cdot 8^1 + A_1 \cdot 8^0$$

При переводе удобно пользоваться таблицей степеней восьмерки:

Таблица 5. Степени числа 8

n (степень)	0	1	2	3	4	5	6
$8^n$	1	8	64	512	4096	32768	262144

Пример . Число  $75013_8$  перевести в десятичную систему счисления.

$$75013_8 = 7 \cdot 8^4 + 5 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 3 \cdot 8^0 = 31243_{10}$$

3. Для перевода шестнадцатеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 16, и вычислить по правилам десятичной арифметики:

$$X_{16} = A_n \cdot 16^{n-1} + A_{n-1} \cdot 16^{n-2} + A_{n-2} \cdot 16^{n-3} + \dots + A_2 \cdot 16^1 + A_1 \cdot 16^0$$

При переводе удобно пользоваться таблицей степеней числа 16:

Таблица 6. Степени числа 16

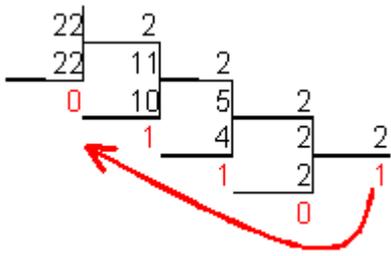
n (степень)	0	1	2	3	4	5	6
$16^n$	1	16	256	4096	65536	1048576	16777216

Пример . Число  $FDA1_{16}$  перевести в десятичную систему счисления.

$$FDA1_{16} = 15 \cdot 16^3 + 13 \cdot 16^2 + 10 \cdot 16^1 + 1 \cdot 16^0 = 64929_{10}$$

4. Для перевода десятичного числа в двоичную систему его необходимо последовательно делить на 2 до тех пор, пока не останется остаток, меньший или равный 1. Число в двоичной системе записывается как последовательность последнего результата деления и остатков от деления в обратном порядке.

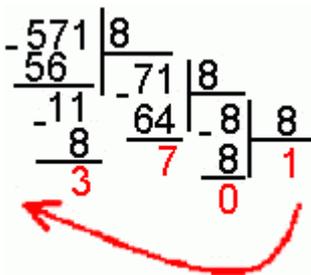
Пример. Число  $22_{10}$  перевести в двоичную систему счисления.



$$22_{10} = 10110_2$$

5. Для перевода десятичного числа в восьмеричную систему его необходимо последовательно делить на 8 до тех пор, пока не останется остаток, меньший или равный 7. Число в восьмеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.

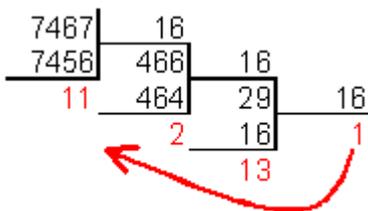
Пример. Число  $571_{10}$  перевести в восьмеричную систему счисления.



$$571_{10} = 1073_8$$

6. Для перевода десятичного числа в шестнадцатеричную систему его необходимо последовательно делить на 16 до тех пор, пока не останется остаток, меньший или равный 15. Число в шестнадцатеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.

Пример. Число  $7467_{10}$  перевести в шестнадцатеричную систему счисления.



$$7467_{10} = 1D2B_{16}$$

7. Чтобы перевести число из двоичной системы в восьмеричную, его нужно разбить на триады (тройки цифр), начиная с младшего разряда, в случае необходимости дополнив старшую триаду нулями, и каждую триаду заменить соответствующей восьмеричной цифрой (табл. 3).

Пример. Число  $1001011_2$  перевести в восьмеричную систему счисления.

$$001\ 001\ 011_2 = 113_8$$

8. Чтобы перевести число из двоичной системы в шестнадцатеричную, его нужно разбить на тетрады (четверки цифр), начиная с младшего разряда, в случае необходимости дополнив старшую тетраду нулями, и каждую тетраду заменить соответствующей восьмеричной цифрой (табл. 3).

Пример. Число  $1011100011_2$  перевести в шестнадцатеричную систему счисления.

$$0010\ 1110\ 0011_2 = 2E3_{16}$$

9. Для перевода восьмеричного числа в двоичную необходимо каждую цифру заменить эквивалентной ей двоичной триадой.

Пример. Число  $531_8$  перевести в двоичную систему счисления.

$$531_8 = 101011001_2$$

10. Для перевода шестнадцатеричного числа в двоичную необходимо каждую цифру заменить эквивалентной ей двоичной тетрадой.

Пример. Число  $EE8_{16}$  перевести в двоичную систему счисления.

$$EE8_{16} = 111011101000_2$$

11. При переходе из восьмеричной системы счисления в шестнадцатеричную и обратно, необходим промежуточный перевод чисел в двоичную систему.

Пример 1. Число  $FEA_{16}$  перевести в восьмеричную систему счисления.

$$FEA_{16} = 111111101010_2$$

$$111\ 111\ 101\ 010_2 = 7752_8$$

Пример 2. Число  $6635_8$  перевести в шестнадцатеричную систему счисления.

$$6635_8 = 110110011101_2$$

$$1101\ 1001\ 1101_2 = D9D_{16}$$

*В цикле «пока» тело цикла выполняется до тех пор, пока выполняется условие.*

## Алгебра логики и доказательство

### Высказывания, операции над высказываниями

**Высказыванием** называется повествовательное предложение, про которое можно сказать истинно оно (обозначается буквой **И** или цифрой **1**), или ложно (обозначается буквой **Л** или цифрой **0**).

#### ПРИМЕР

«земля плоская» – данное высказывание является ложным (**Л**);

«5 – простое число» – данное высказывание является истинным (**И**);

« $x > 6$ » – данное предложение не является высказыванием, т. к. оно содержит переменную и может принимать значение **И** при одних значениях переменной (например,  $x=7$  и др.) и значение **Л** при других значениях переменной (например,  $x=5$  и др.);

«Является ли 10 четным числом?» – данное предложение не является высказыванием, т. к. оно не является повествовательным.

Каждое из высказываний можно обозначить своей буквой. Пусть, например, **P** обозначает высказывание «земля плоская», **S** обозначает высказывание «5 – простое число».

Из простых высказываний с помощью логических операций можно построить составные высказывания.

Обычно рассматривают следующие логические операции (операции над высказываниями):

Название	Прочтение	Обозначение
Отрицание (инверсия)	<b>Не</b>	$\neg$
Конъюнкция	<b>И</b>	$\&, \cdot, \wedge$
Дизъюнкция	<b>Или</b>	$+, \vee$
Строгая дизъюнкция	<b>Либо ..., либо...</b>	$\oplus, \veebar, \text{XOR}$
Импликация	<b>Если..., то...</b>	$\rightarrow$
Эквивалентность	<b>Тогда и только тогда</b>	$\leftrightarrow, \sim$

Чтобы уметь определять является составное высказывание истинным или ложным, необходимо разобраться со смыслом логических операций, т. е. какой эффект они оказывают на истинностное значение простых высказываний. Это можно сделать с помощью так называемых *таблиц истинности*.

- **Отрицанием** (инверсией от лат. *переворачивание*) произвольного высказывания **P** называется высказывание  $\neg P$  (Читается: «не P», или «**неверно, что P**»), которое истинно, когда высказывание P ложно, и ложно, когда высказывание P истинно.

Вместо обозначения  $\neg P$  можно использовать обозначение  $\bar{P}$ .  
 Определяющая таблица истинности отрицания высказывания приведена в таблице.

P	$\bar{P}$
0	1
1	0

### ПРИМЕР

Возьмем два простых высказывания: **P** – «Вася выучил теорему», **Q** – «Он получил пятерку». С помощью **отрицания** составим новые высказывания:  $\neg P$  – «Вася **не** выучил теорему»,  $\neg Q$  – «Он **не** получил пятерку».

Высказывание  $\neg P$  – «Вася **не** выучил теорему», будет истинно, если высказывание P – «Вася выучил теорему» – ложно, и наоборот.

•**Конъюнкцией** или **логическим умножением** двух высказываний **P** и **Q** называется составное высказывание  **$P \wedge Q$**  (Читается: «**P И Q**»), которое принимает истинное значение только в том случае, когда истинны обе его составные части.

Такое определение хорошо согласуется с обычным пониманием союза «**и**» в разговорном языке.

Определяющая таблица истинности:

P	Q	$P \wedge Q$
0	0	0
0	1	0
1	0	0
1	1	1

#### ПРИМЕР

Возьмем те же простые высказывания **P** – «Вася выучил теорему» и **Q** – «Он получил пятерку». С помощью *конъюнкции* составим новое высказывание:  **$P \wedge Q$**  – «Вася выучил теорему **И** он получил пятерку».

Высказывание  **$P \wedge Q$**  – «Вася выучил теорему **И** он получил пятерку» будет истинно, если высказывания **P** – «Вася выучил теорему» и **Q** – «Он получил пятерку» будут истинны одновременно (См. последнюю строку таблицы), если же хотя бы одно из высказываний **P** или **Q** ложно, то ложно и высказывание  **$P \wedge Q$** .

•**Дизъюнкцией** или **логическим сложением** двух высказываний **P** и **Q** называется составное высказывание  **$P \vee Q$**  (Читается: «**P ИЛИ Q**»), которое принимает истинное значение, если хотя бы одна из ее составных частей имеет истинное значение.

Другими словами, истинность  **$P \vee Q$**  означает, что истинно или **P**, или **Q**, или и то, и другое. Таблица истинности дизъюнкции:

P	Q	$P \vee Q$
0	0	0
0	1	1
1	0	1
1	1	1

## ПРИМЕР

Возьмем те же высказывания  $P$  и  $Q$ . С помощью *дизъюнкции* составим новое высказывание:  $P \vee Q$  – «Вася выучил теорему **ИЛИ** получил пятерку».

Высказывание  $P \vee Q$  – «Вася выучил теорему **или** он получил пятерку» будет истинно, если хотя бы одно из высказываний  $P$  или  $Q$  будет истинно (См. выделенные строки таблицы), если же оба высказывания  $P$  и  $Q$  ложны, то ложно и высказывание  $P \vee Q$ .

•*Строгой дизъюнкцией* двух высказываний  $P$  и  $Q$  называется составное высказывание  $P \vee\vee Q$  (Читается: «**либо**  $P$  **либо**  $Q$ »). которое принимает истинное значение, если только одна из ее составных частей имеет истинное значение.

Таблица истинности строгой дизъюнкции:

$P$	$Q$	$P \vee\vee Q$
0	0	0
0	1	1
1	0	1
1	1	0

## ПРИМЕР

Возьмем те же высказывания  $P$  и  $Q$ . С помощью *строгой дизъюнкции* составим новое высказывание:  $P \vee\vee Q$  – «**Либо** Вася выучил теорему, **либо** получил пятерку».

Высказывание  $P \vee\vee Q$  – «**Либо** Вася выучил теорему, **либо** получил пятерку» будет истинно, если только одно из высказываний  $P$  или  $Q$  будет истинно, если же оба высказывания  $P$  и  $Q$  ложны или оба истинны, то ложно и высказывание  $P \vee\vee Q$ .

•*Импликацией* двух высказываний  $P$  и  $Q$  называется составное высказывание  $P \rightarrow Q$  (Читается: «**Если**  $P$ , **то**  $Q$ », или «из  $P$ , **следует**  $Q$ », или « $P$  **влечет**  $Q$ », или « $P$  **достаточно для**  $Q$ », или « $Q$  **необходимо для**  $P$ »), которое принимает ложное значение, если высказывание  $P$  истинно, а высказывание  $Q$  ложно. Другими  $P \rightarrow Q$  словами ложно, только если из истины следует ложь.

Сложное высказывание, построенное с помощью импликации, называется *условным*. При этом **P** называется *предпосылкой*, а **Q** – *заключением*.

Таблица истинности импликации:

P	Q	$P \rightarrow Q$
0	0	1
0	1	1
1	0	0
1	1	1

### ПРИМЕР

Возьмем те же высказывания **P** и **Q**. С помощью *импликации* составим новое высказывание:  $P \rightarrow Q$  – «Если Вася выучил теорему, **то** он получил пятерку», или «из того что Вася выучил теорему, **следует** что он получил пятерку».

Высказывание  $P \rightarrow Q$  – «Если Вася выучил теорему, **то** он получил пятерку» будет ложно только если предпосылка **P** будет истинна, а заключение **Q** – ложно.

Если высказывание  $P \rightarrow Q$  истинно, то говорят, что **P** *логически следует* из **Q**, обозначается  $P \vDash Q$ .

•*Эквиваленцией* двух высказываний **P** и **Q** называется составное высказывание  $P \leftrightarrow Q$  (Читается: «**P** **тогда и только тогда, когда** **Q**»), которое принимает истинное значение, тогда и только тогда, когда оба высказывания истинны, или оба высказывания ложны.

Таблица истинности строгой дизъюнкции:

P	Q	$P \leftrightarrow Q$
0	0	1
0	1	0
1	0	0
1	1	1

С помощью *эквиваленции* составим новое высказывание:  $P \leftrightarrow Q$  – «Вася выучил теорему **тогда и только тогда, когда** он получил пятерку»

Два составных высказывания, построенные из одних и тех же простых утверждений, но разными путями, могут принимать одинаковые значения истинности на любом возможном наборе значений истинности своих составных частей. Такие высказывания называются *логически эквивалентными*. Логическая эквивалентность высказываний  $P$  и  $Q$  обозначается  $P \Leftrightarrow Q$  или  $P = Q$ .

### ПОСЛЕДОВАТЕЛЬНОСТЬ ОПЕРАЦИЙ

1. Инверсия;
2. Конъюнкция;
3. Дизъюнкция;
4. Импликация и эквивалентность;

Порядок можно изменить с помощью скобок.

### ПРИМЕР

Высказывание  $\neg(P \wedge (\neg Q))$  логически эквивалентно утверждению  $\neg P \vee Q$ . Для того, чтобы это доказать, составим совместную таблицу истинности для составных высказываний:

Вспомогательные колонки используются для построения обоих выражений из  $P$  и  $Q$ .

$P$	$Q$	$\neg P$	$\neg Q$	$P \wedge (\neg Q)$	$\neg(P \wedge (\neg Q))$	$\neg P \vee Q$
л	л	и	и	л	и	и
л	и	и	л	л	и	и
и	л	л	и	и	л	л
и	и	л	л	л	и	и

Две последние колонки таблицы идентичны. Это означает, что высказывание  $\neg(P \wedge (\neg Q))$  логически эквивалентно высказыванию  $\neg P \vee Q$ , т.е.  $\neg(P \wedge (\neg Q)) = \neg P \vee Q$ .

Высказывание  $(\neg Q) \rightarrow (\neg P)$

называется *противоположным* или *контрапозитивным* к высказыванию

$P \rightarrow Q$ .

### ПРИМЕР

Показать, что высказывание  $(\neg Q) \rightarrow (\neg P)$  логически эквивалентно высказыванию  $P \rightarrow Q$ , т.е.  $(\neg Q) \rightarrow (\neg P) = P \rightarrow Q$ .

### РЕШЕНИЕ

Рассмотрим совместную таблицу истинности.

P	Q	$P \rightarrow Q$	$\neg P$	$\neg Q$	$(\neg Q) \rightarrow (\neg P)$
л	л	и	и	и	и
л	и	и	и	л	и
и	л	л	л	и	л
и	и	и	л	л	и

Поскольку два последних столбца этой таблицы совпадают, то и высказывания, о которых идет речь, логически эквивалентны.

## Логические законы

### **Коммутативность**

$$A \wedge B = B \wedge A \quad A \vee B = B \vee A$$

Коммутативность означает, что можно менять местами члены конъюнкции (дизъюнкции).

### **Ассоциативность**

$$(A \wedge B) \wedge C = A \wedge (B \wedge C) = A \wedge B \wedge C$$

$$(A \vee B) \vee C = A \vee (B \vee C) = A \vee B \vee C$$

Ассоциативность означает, что можно не ставить скобки при нескольких последовательных применениях конъюнкции (дизъюнкции) или наоборот ставить их, так как удобно.

### **Дистрибутивность**

$$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$$

$$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$$

Дистрибутивность означает, что можно раскрывать скобки при применениях конъюнкции к дизъюнкции и наоборот.

Замечание: дистрибутивный закон легче запомнить, если использовать другие обозначения конъюнкции ( $\cdot$ ) и дизъюнкции ( $+$ ),

$$\text{Вместо } A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$$

$$\text{получим } A \cdot (B + C) = (A \cdot B) + (A \cdot C).$$

### **Идемпоентность**

$$A \wedge A = A$$

$$A \vee A = A$$

### **Закон исключенного третьего**

$$A \vee \bar{A} = 1$$

### **Закон противоречия**

$$A \wedge \bar{A} = 0$$

### **Закон двойного отрицания**

$$\bar{\bar{A}} = A$$

### **Законы действий с константами**

$$A \vee 1 = 1 \quad A \wedge 1 = A$$

$$A \vee 0 = A \quad A \wedge 0 = 0$$

*Закон Де Моргана*

$$\overline{A \wedge B} = \overline{A} \vee \overline{B}$$

$$\overline{A \vee B} = \overline{A} \wedge \overline{B}$$

Доказать справедливость каждого из законов, значит доказать логическую эквивалентность левой и правой частей этих равенств, например, докажем закон Де Моргана:

A	B	$A \wedge B$	$\overline{A \wedge B}$	$\overline{A}$	$\overline{B}$	$\overline{A} \vee \overline{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

## Предикаты и кванторы

Логика высказываний применяется к простым высказываниям, которые либо истинны, либо ложны. Утверждения, содержащие одну и более переменных, могут быть верными при некоторых значениях переменных и ложными при других.

**Предикатом** называется утверждение, содержащее переменные и принимающее значение истины или лжи в зависимости от значений переменных.

### ПРИМЕР

Утверждение « $x$  — целое число, удовлетворяющее соотношению  $x = x^2$ » является предикатом, поскольку оно истинно при  $x=0$  или  $x=1$  и ложно в любом другом случае.

Логические операции можно применять и к предикатам.

В общем случае истинность составного предиката в конечном счете зависит от значений, входящих в него переменных. Однако существуют логические операторы (называемые кванторами), применение которых к предикатам превращает последние в ложные или истинные высказывания.

### ПРИМЕР

Какие из следующих высказываний истинны, а какие ложны?

1. Сумма внутренних углов любого треугольника равна  $180^\circ$ .
2. У всех кошек есть хвост.
3. Найдется целое число  $x$ , удовлетворяющее соотношению  $x = 2$ .
4. Существует простое четное число.

### РЕШЕНИЕ

*1. Истинно.*

*2. Ложно. У бесхвостой кошки хвоста нет.*

*3. Ложно.*

*4. Истинно. Число 2 является и простым, и четным.*

В этом примере мы имеем дело с набором объектов и утверждениями о том, что некоторое свойство имеет место для всех рассматриваемых объектов, или что найдется (существует) по крайней мере, один объект, обладающий данным свойством.

$\exists$  - «существует»  
 $\forall$  - «принадлежит»

Выражения «для всех» и «найется» («существует») называются кванторами и обозначаются, соответственно,  $\forall$  и  $\exists$ . Включая в предикат кванторы, мы превращаем его в высказывание. Поэтому предикат с кванторами может быть истинным или ложным.

### ПРИМЕР

Обозначим через  $P(x)$  предикат « $x$  — целое число и  $x^2 = 16$ ».

Высказывание  $\exists x: P(x)$  означает, что **найется** целое число  $x$ , удовлетворяющее уравнению  $x^2 = 16$ . Высказывание, конечно, истинно, поскольку уравнение  $x^2 = 16$  превращается в верное тождество при  $x = 4$ . Кроме того,  $x = -4$  — также решение данного уравнения. Однако нам не требуется рассуждать о знаке переменной  $x$ , чтобы проверить истинность высказывания  $\exists x: P(x)$ .

### ПРИМЕР

Пусть  $P(x)$  — предикат: « $x$  — вещественное число и  $x^2+1=0$ ». Выразите словами высказывание:  $\exists x: P(x)$  и определите его истинностное значение.

### РЕШЕНИЕ

Данное высказывание можно прочитать так: **существует** действительное число  $x$ , удовлетворяющее уравнению  $x^2+1=0$ . Поскольку квадрат любого действительного числа неотрицателен, т. е.  $x^2 \geq 0$ , мы получаем, что  $x^2+1 \neq 0$ . Следовательно, утверждение  $\exists x: P(x)$  ложно.

Отрицание высказывания  $\exists x: P(x)$  записывается в следующем виде:  $\neg(\exists x: P(x))$ . Это, естественно, истинное высказывание, которое означает, что **не существует** действительного числа  $x$ , удовлетворяющего условию  $x^2+1=0$ . Иными словами, каково бы ни было действительное  $x$ ,  $x^2+1 \neq 0$ . В символьной форме это можно записать как  $\forall x: \neg P(x)$ .

Для общего предиката  $P(x)$  есть следующие логические эквивалентности:

$$\neg(\exists x: P(x)) \Leftrightarrow \forall x: \neg P(x);$$

$$\neg(\forall x: P(x)) \Leftrightarrow \exists x: \neg P(x).$$

Как показывает следующий пример, некоторые трудности возникают, когда в высказывании участвует более одного квантора.

### ПРИМЕР

Предположим, что  $x$  и  $y$  действительные числа, а  $P(x, y)$  обозначает предикат  $x + y = 0$ . Выразите каждое из высказываний словами и определите их истинность.

а)  $\forall x \forall y: P(x, y)$ ;

б)  $\exists y: \forall x P(x, y)$ .

### РЕШЕНИЕ

$\forall x \forall y: P(x, y)$  говорит о том, что **для любого** действительного числа  $x$  **найдется** такое действительное число  $y$ , что  $x + y = 0$ . Оно, очевидно, верно, поскольку какое бы число  $x$  мы ни взяли, число  $y = -x$  обращает равенство  $x + y = 0$  в верное тождество.

(б) Высказывание  $\exists y: \forall x P(x, y)$  читается следующим образом: **существует** такое действительное число  $y$ , что для любого действительного числа  $x$  выполнено равенство  $x + y = 0$ . Это, конечно, не так: не существует вещественного числа  $y$ , обладающего указанным свойством. Следовательно, высказывание ложно.

## Логические и битовые операции

В компьютерах основной единицей информации является *бит*. Бит принимает два возможных значения 0 или 1, иными словами, бит-одноразрядное двоичное число. Для обозначения значения истинности высказывания мы также использовали 0, 1. Таким образом, если  $A$  - высказывание, то его значение истинности один бит информации. Переменная, принимающая значения во множестве  $\{0,1\}$ , обычно называется булевой переменной. Т. е. булева переменная — это такая переменная, задание значения которой определяет один бит информации.

Компьютерные битовые (или логические) операции соответствуют операциям над высказываниями (вернее, над их значениями истинности).

Приведем таблицы, определяющие операции NOT, OR ( $\cup$ ), AND ( $\cap$ ), XOR ( $\oplus$ ):

	NOT
0	1
1	0

OR	0	1
0	0	1
1	1	1

AND	0	1
0	0	0
1	0	1

XOR	0	1
0	0	1
1	1	0

**Битовой строкой длины  $n$**  называется последовательность длины  $n$ , элементами которой являются биты.

Например, 0110010 — битовая строка длины 7. Битовые операции естественным образом (поэлементно) распространяются на битовые строки равной длины bitwise (пер. побитовое).

Их обозначения: bitwise NOT, bitwise OR, bitwise AND, bitwise XOR.

### ПРИМЕР

bitwise NOT (0110010) = 1001101.

(0110010) bitwise OR (0010111) = (0110111).

## Классификация моделей данных

**Модель данных** – это набор правил, по которым организуются данные. Это очень простое определение можно уточнить.

**Модель данных** – это некоторая абстракция, которая, будучи приложена к конкретным данным, позволяет пользователям и разработчикам трактовать их как информацию, то есть сведения, содержащие не только данные, но и взаимосвязи между ними.

Классификация моделей, их описание появились после разработки реляционной модели. До этого разрабатывали БД, используя имеющиеся технологии. И значительно позднее проанализировали существующие базы данных и выполнили их теоретическое описание.

Теоретико-графовые модели отражают совокупность объектов реального мира в виде графа. В зависимости от типа графа различают иерархическую и сетевую модели. Иерархическая и сетевая модели данных стали применяться в СУБД в начале 60-х годов 20 века. В настоящее время они используются реже, чем реляционная модель данных.

Для работы со сложными наборами данных математики разработали иерархическую модель данных. Эта модель появилась раньше других даталогических моделей. Именно эта модель данных использована в первой официально признанной промышленной СУБД фирмы ИВМ.

Принято выделять три группы моделей данных: инфологические, даталогические и физические.

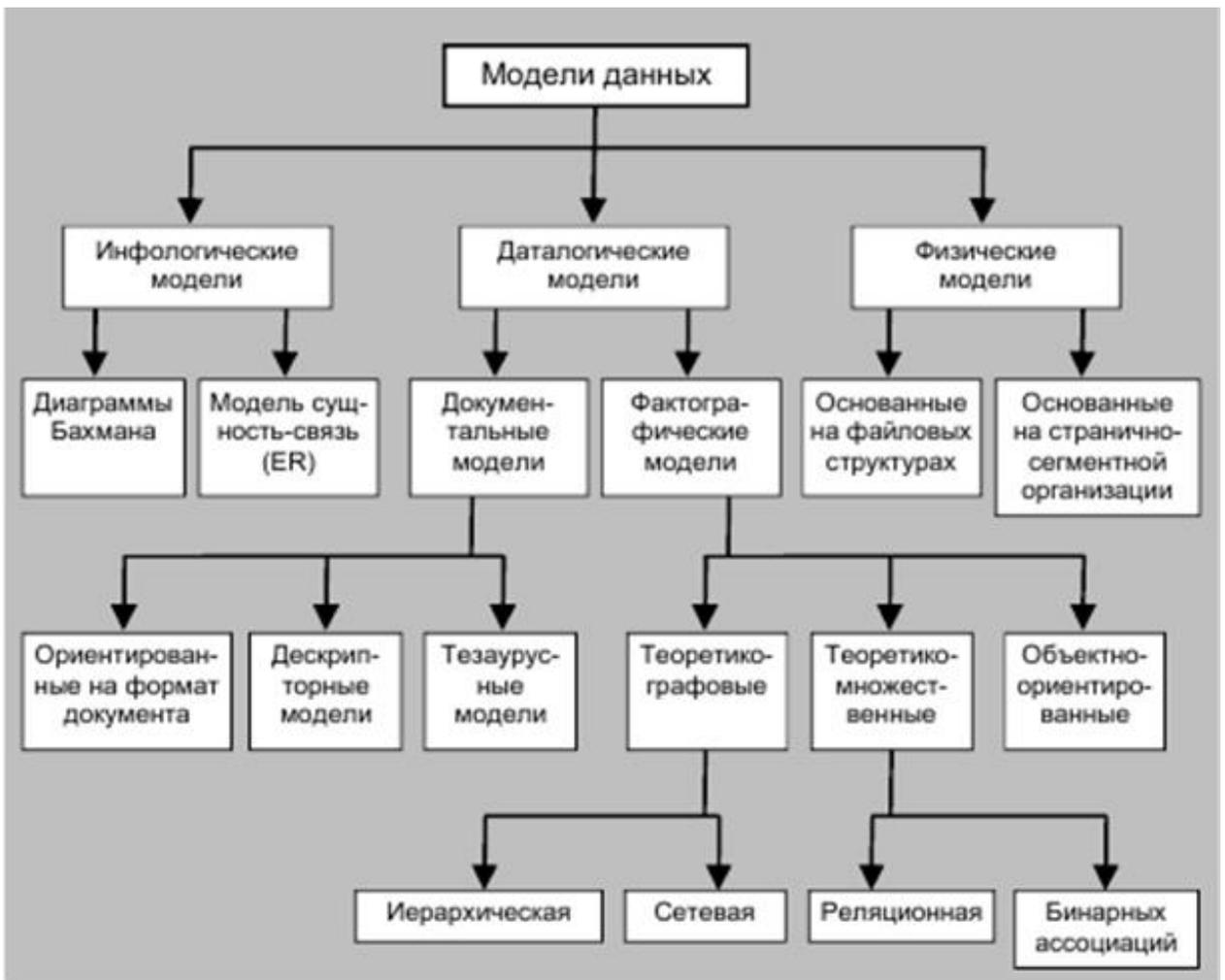


Рис.1 Модели данных

### Инфологические модели

*Инфологическая* (семантическая) модель – это обобщённое, не привязанное к какой-либо ЭВМ и СУБД описание предметной области. Это описание, выполненное с использованием естественного языка, математических формул, таблиц, графиков и других средств объединяет частные представления о содержимом базы данных, полученные в результате опроса пользователей, и представления разработчиков о данных, которые могут потребоваться в будущих приложениях.

Такая человеко-ориентированная модель полностью независима от физических параметров среды хранения данных. Поэтому инфологическая модель не должна изменяться до тех пор, пока она адекватно отражает предметную область, то есть до тех пор, пока не произошли изменения в предметной области.

Из инфологических моделей далее рассматривается получившая наибольшее распространение модель «сущность-связь», предложенная Питером Ченом.

**Физическая модель** – это модель, создаваемая путём замены объектов моделирующими устройствами, которые имитируют определённые характеристики либо свойства этих объектов. При этом моделирующее устройство имеет ту же качественную природу, что и моделируемый объект.

Физическая модель представляет собой аналоговую модель, в которой между параметрами объекта и модели одинаковой физической природы существует однозначное соответствие.

### Даталогические модели

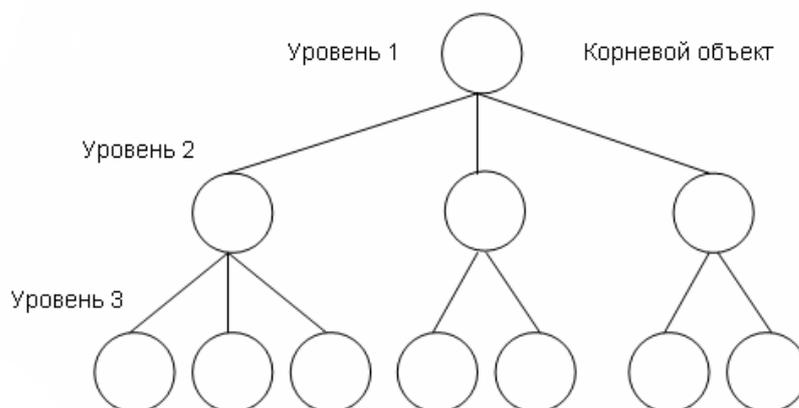
*Даталогические* модели являются компьютерно-ориентированными, они поддерживаются конкретными СУБД. С их помощью СУБД даёт возможность пользователям осуществлять доступ к хранимым данным, не заботясь об их физическом расположении. Так как доступ к данным осуществляется с помощью конкретной СУБД, то даталогические модели описываются на *языке описания данных* (ЯОД) используемой СУБД.

К этой группе относятся такие широко известные модели как иерархическая, сетевая, реляционная.

### Иерархическая модель

Иерархическая модель предполагает хранение данных в виде, похожем на организацию каталогов в MS DOS: все каталоги начинаются с корневого и ветвятся подобно дереву. К каждому файлу есть только один путь, то есть файлу соответствует одно имя каталога.

**Иерархическая модель** представляет собой совокупность элементов, расположенных в порядке их подчинения от общего к частному и образующих перевернутое по структуре дерево (граф).

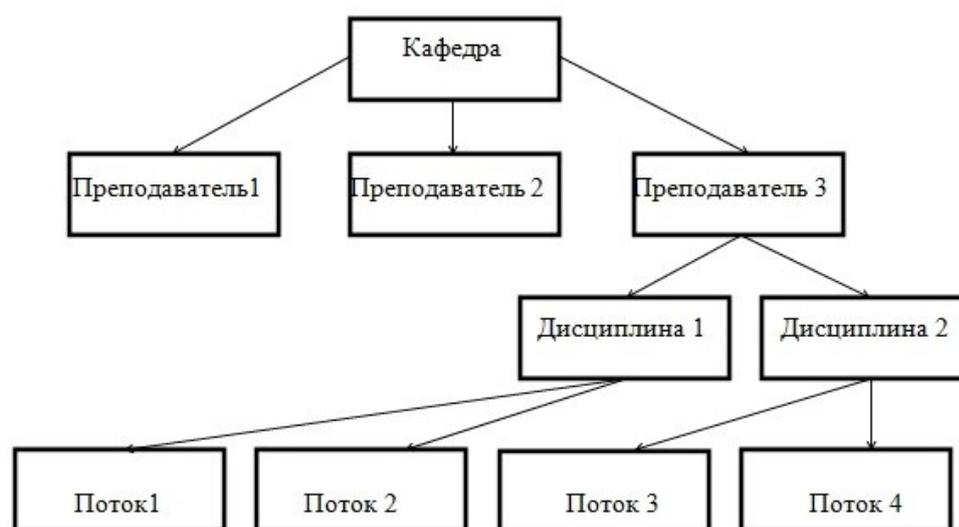


К основным понятиям иерархической структуры относятся уровень, узел и связь.

Узел - это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне.

Иерархическое дерево имеет только одну вершину, не подчиненную никакой другой вершине и находящуюся на самом верхнем - первом уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т. д. уровнях.

Количество деревьев в базе данных определяется числом корневых записей. К каждой записи базы данных существует только один иерархический путь от корневой записи.



Пример иерархической модели

В реальном мире некоторые объекты по своей сути составляют иерархические структуры: одни объекты являются родительскими, другие – дочерними. Иерархия проста и естественна для отображения взаимосвязей между объектами. Достаточно вспомнить многочисленные классификации, используемые в разных областях знаний, например, приведённую выше классификацию моделей данных. В качестве другого примера можно привести структуру данных предприятия.

В иерархической БД все записи ветвятся от одной корневой. Запись имеет всегда только одного родителя и сама тоже может быть родителем для другой записи.

Главное достоинство иерархической модели – скорость. Поскольку все отношения между таблицами predeterminedены и являются статическими, поисковые и другие операции над набором данных выполняются очень быстро.

Наиболее существенный недостаток – негибкость. Поскольку отношения хранятся внутри каждой записи, данные имеют смысл только в определённом контексте. Другой недостаток – трудность переноса данных с компьютера на компьютер. Третий недостаток заключается в том, что глобальные изменения данных практически невозможны. При изменении требуется, чтобы каждая запись, включая родительские и дочерние, была модифицирована индивидуально.

Работа с этой моделью данных предполагает значительный объём знаний. Большинство БД, использующих иерархическую модель, требует специально подготовленного персонала для обеспечения правильного функционирования.

## Сетевая модель

Сетевая модель предложена для обеспечения гибкости в управлении данными. На разработку этой модели большое влияние оказал американский ученый Ч.Бахман.

Основные принципы сетевой модели данных были сформулированы в середине 60-х годов. Эталонный вариант сетевой модели данных описан в отчетах рабочей группы по языкам баз данных CODASYL (COncference on DAta SYstem Languages) в середине 70-х годов.

Сетевая модель отличается от иерархической тем, что позволяет определять для записи более чем одно групповое отношение. Эта модели состоит из множества записей, которые могут быть владельцами или членами групповых отношений. Сетевая модель позволяет производить поиск в различных структурах и поддерживает для записей отношение «одна ко многим».

Как и в иерархической БД, информация о связях хранится в записях и должна быть предопределена. Поэтому сетевая модель данных имеет те же ограничения, что и иерархическая.

В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.



Пример сетевой модели

## Реляционная модель

Сложность практического использования иерархических и сетевых СУБД, желание пользователей оперировать более крупными объектами, чем элементы данных заставили искать иные способы представления данных и послужило причиной возникновения новой структуры данных – реляционной (табличной). Работа с таблицами понятна и привычна каждому пользователю. Создателем реляционной модели является математик, сотрудник фирмы IBM Э.Ф. Кодд (1970 г.). Он же ввел два языка манипулирования данными SQL и QBE.

**Реляционная модель данных** объекты и связи между ними представляет в виде таблиц, при этом связи тоже рассматриваются как объекты. Все строки, составляющие таблицу в реляционной базе данных, должны иметь первичный ключ. Все современные средства СУБД поддерживают реляционную модель данных.

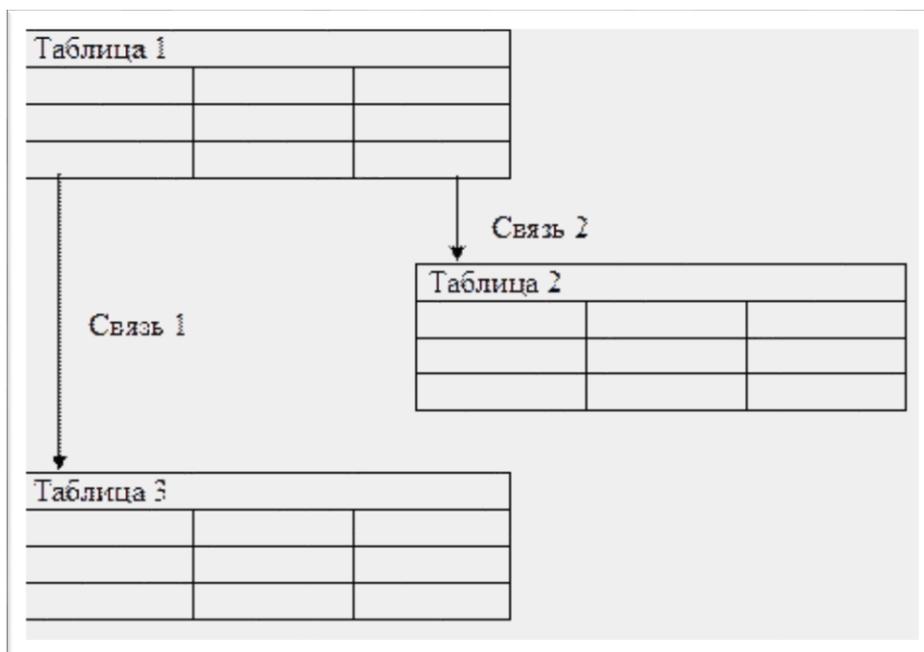


Рис. Реляционная модель данных

Эта модель характеризуется простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Каждая реляционная таблица представляет собой двумерный массив и обладает **следующими свойствами**:

1. Каждый элемент таблицы соответствует одному элементу данных.
2. Все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип и длину.

3. Каждый столбец имеет уникальное имя.
4. Одинаковые строки в таблице отсутствуют;
5. Порядок следования строк и столбцов может быть произвольным.

### Что понимается под верификацией модели.

Что такое верификация? Описание и определение понятия Верификация-это (лат., от *verus* – истинный, *facio* – делаю) способ подтверждения, проверяемость, эмпирическое подтверждение теоретических алгоритмов, положений, процедур или программ, сопоставляя их с эталонными, эмпирическими, опытными данными, программами или алгоритмами. Верификация обозначает также соответствие предопределенным эталонным характеристикам конечного продукта. Термин «верификация» используют для обозначения методики распознавания искажения, укрывательства, лжи. Такое разное толкование этого термина объясняется широкими возможностями проверить соответствие различных характеристик предъявленным требованиям к ним.

1. Термин «верифицировано» применяется, обозначая соответствующий статус.

2. Деятельность по подтверждению может включать: осуществлять альтернативные расчеты сравнивать научную и техническую документации по новому проекту с аналогичной документацией по апробированному проекту проводить испытания и демонстрации анализировать документы до их выпуска.

Верификацию легко спутать с валидацией. Верификация всегда будет опираться на сравнение реальных опытных образцов с эталонными, которые создаются на фазе проектирования. Термины «верификация» и «валидация» зачастую используют в технической литературе. Валидацией и верификацией можно назвать виды деятельности, направленные на проведение контроля качества программного продукта для обнаружения на стадиях разработки ошибок в нем. Кажется, они имеют общую цель. Однако они обладают различиями в проверяемых свойствах, ограничениях и правилах, не соблюдая которые можно назвать ошибкой.

Верификация – это проверка на соответствие программного обеспечения технической документации, которая представлена техзаданием, архитектурой или моделью предметной области. В задачи верификации включается и сопоставление процедуры расчетов с процессом их разработки, правилами и стандартами. Верификация данных должна выполняться, чтобы установить функционирования программы в соответствии с установленными нормами, требованиями, проектными решениями и пользовательской документацией.

В отличие от верификации валидация подразумевает проверку на соответствие разрабатываемых или сопровождаемых программных продуктов требований заказчиков или пользователей. Эти потребности зачастую не отражаются ни в какой документации. Именно поэтому валидация является менее формализованной, чем верификация. В этом процессе участвуют представитель заказчика, пользователя, а может находиться аналитик или эксперт в предметной области. Иными словами, те, которые могут представлять конкретные потребности и реальные нужды этих лиц.

## Этапы информационного моделирования



### 1 Этап:

#### 1.1. Описание задачи.

Задача формулируется на обычном языке. По характеру постановки все задачи можно разделить на две основные группы. К первой группе относятся задачи в которых требуется исследовать, как изменяется характеристика объекта при некотором воздействии на него, «*что будет, если...?*»

К второй группе относятся задачи в которых требуется определить, какое воздействие надо произвести на объект, чтобы его параметры удовлетворяли некоторому заданному воздействию, «*как сделать чтобы ...?*»

#### 1.2. Определение целей моделирования. Они таковы:

- *понимание* – модель нужна для того, чтобы понять, как устроен конкретный объект, какова его структура, основные свойства, законы развития и взаимодействие с окружающим миром;

- *управление* – модель нужна, чтобы научиться управлять объектом (или процессом) и определить наилучшие способы управления при заданных целях и критериях;

- *прогнозирование* – модель нужна для того, чтобы научиться прогнозировать прямые и косвенные последствия реализации заданных способов и форм воздействия на объект.

Ответить на вопросы:

- Что будет, если...? – цель, определение последствий воздействия на объект и принятия правильного решения.

- Как сделать, чтобы...? – цель, создание объекта с заданными свойствами.

## 2 Этап:

Формализация задачи связана с созданием *формализованной модели*, то есть записанной на каком-либо формальном языке.

Для решения задачи на компьютере больше всего подходит язык математики. В такой модели связь между исходными данными и конечными результатами фиксируются с помощью различных формул и накладываются ограничения на допустимые значения параметров.

**Происходит ранжирование** – разделение входных параметров по степени важности влияния их изменений на выходные.

Выбор наиболее существенной информации при создании модели и ее сложность обусловлены целью моделирования.

## 3 Этап:

Поиск *математического описания*, разработка алгоритмов, выбор программного средства и составление программы для ЭВМ.

**Компьютерная модель** – модель, реализованная средствами программной среды.

## 4 Этап:

Тестирование программы, исправление ошибок, численный эксперимент.

**Тестирование** – процесс проверки правильности модели.

**Тест** – набор исходных данных, для которых заранее известен результат.

## Основные этапы решения задач на компьютере.

Процесс решения задач на компьютере — это совместная деятельность человека и ЭВМ. Этот процесс можно представить в виде нескольких последовательных этапов. На долю человека приходятся этапы, связанные с творческой деятельностью — постановкой, алгоритмизацией, программированием задач и анализом результатов, а на долю компьютера - этапы обработки информации в соответствии с разработанным алгоритмом.

*Рассмотрим эти этапы на следующем примере: пусть требуется вычислить сумму двух целых чисел и вывести на экран видеомонитора результат.*

**Первый этап - постановка задачи.** На этом этапе участвует человек, хорошо представляющий предметную область задачи. Он должен четко определить цель задачи, дать словесное описание содержания задачи и предложить общий подход к ее решению.

*Для задачи вычисления суммы двух целых чисел человек, знающий, как складываются числа, может описать задачу следующим образом: ввести два целых числа, сложить их и вывести сумму в качестве результата решения задачи.*

**Второй этап - математическое или информационное моделирование.** Цель этого этапа - создать такую математическую модель решаемой задачи, которая может быть реализована в компьютере. Существует целый ряд задач, где математическая постановка сводится к простому перечислению формул и логических условий. Этот этап тесно связан с первым этапом, и его можно отдельно не рассматривать, однако возможно, что для полученной модели известны несколько методов решения, и тогда предстоит выбрать лучший.

*Для вышеописанной задачи данный этап сведется к следующему: введенные в компьютер числа започнем в памяти под именами  $A$  и  $B$ , затем вычислим значение суммы этих чисел по формуле  $A+B$ , и результат започнем в памяти под именем  $Summa$ .*

**Третий этап — алгоритмизация задачи.** На основе математического описания необходимо разработать алгоритм решения.

Алгоритмом	<i>называется точное предписание, определяющее последовательность действий исполнителя, направленных на решение поставленной задачи. В роли исполнителей алгоритмов могут выступать люди, роботы, компьютеры.</i>
Алгоритм	<i>- формальное однозначное описание последовательности действий.</i>
Алгоритм	<i>- точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату. [ГОСТ 19.004-80].</i>

Разработка алгоритма решения задачи называется **алгоритмизацией**. В процессе алгоритмизации задача сводится к построению последовательности шагов, расположенных в определенном порядке.

**Порядок решения задач на компьютере (пример):**



## Способы записи алгоритмов.

К изобразительным средствам описания алгоритмов относятся следующие основные способы их представления:

- а) словесный (записи на естественном языке);
- б) структурно–стилизированный (записи на языке псевдокода);
- в) программный (тексты на языках программирования);
- г) графический (схемы графических символов).

**Словесный способ записи алгоритмов** представляет собой описание последовательных этапов обработки данных и задается в произвольном изложении на естественном языке. Способ основан на использовании общепринятых средств общения между людьми и, с точки зрения написания, трудностей для авторов алгоритмов не представляет. Однако для “исполнителей” такие описания алгоритмов часто неприемлемы. Они строго не формализуемы, страдают многословностью записей, допускают неоднозначность толкования отдельных предписаний. Поэтому такой способ описания алгоритмов не имеет широкого распространения.

**Структурно-стилизированный способ записи алгоритмов** основан на формализованном представлении предписаний, задаваемых путем использования ограниченного набора типовых синтаксических конструкций. Такие средства описания алгоритмов часто называются *псевдокодами*.

Разновидностью структурно–стилизованного способа описания алгоритмов является алгоритмический язык в русской нотации (АЯРН).

**Программный способ записи алгоритмов** – это алгоритм, записанный на языке программирования, позволяющем на основе строго определенных правил формировать последовательность предписаний, однозначно отражающих смысл и содержание частей алгоритма с целью их последующего исполнения на ЭВМ.

Для **графического изображения алгоритмов** используются *графические символы*. Наиболее распространенными являются блочные символы (блоки), соединяемые линиями передач управления. Существует государственный стандарт на выполнение графической записи алгоритма. Графическая запись алгоритма является наиболее наглядной. Перечень условных графических символов, их наименования, форма, размеры и отображаемые функции устанавливаются ГОСТ 19.003–80.

### Основные свойства алгоритма.

#### 1. **Массовость.**

Алгоритм имеет некоторое число входных величин - аргументов, задаваемых до начала исполнения. Цель выполнения алгоритма — получение результата (результатов), имеющего вполне определенное отношение к исходным данным. Алгоритм указывает последовательность действий по переработке исходных данных в результаты. Для алгоритма можно выбирать различные наборы входных данных из множества допустимых для этого процесса данных, т.е. можно применять алгоритм для решения целого класса задач одного типа, различающихся исходными данными. Это свойство алгоритма обычно называют *массовостью*. Однако существуют алгоритмы, применимые только к единственному набору данных. Можно сказать, что для каждого алгоритма существует свой класс объектов, допустимых в качестве исходных данных. Тогда свойство *массовости* означает применимость алгоритма ко всем объектам этого класса.

#### 2. **Понятность.**

Чтобы алгоритм можно было выполнить, он должен быть понятен исполнителю. *Понятность алгоритма* означает знание исполнителя о том, что надо делать для исполнения этого алгоритма.

#### 3. **Дискретность.**

Алгоритм представляется в виде конечной последовательности шагов (алгоритм имеет *дискретную* структуру) и его исполнение расчленяется на выполнение отдельных шагов (выполнение очередного шага начинается после завершения предыдущего).

#### 4. **Конечность.**

Выполнение алгоритма заканчивается после выполнения *конечного числа шагов*. При выполнении алгоритма некоторые его шаги могут

повторяться многократно. В математике существуют вычислительные процедуры, имеющие алгоритмический характер, но *не* обладающие свойством *конечности*.

#### 5. *Определенность.*

Каждый шаг алгоритма должен быть *четко и недвусмысленно определен* и не должен допускать произвольной трактовки исполнителем. Следовательно, алгоритм рассчитан на *чисто механическое исполнение*. Именно *определенность* алгоритма дает возможность поручить его исполнение *автомату*.

#### 6. *Эффективность.*

Каждый шаг алгоритма должен быть выполнен точно и за конечное время. В этом смысле говорят, что алгоритм должен быть *эффективным*, т.е. действия исполнителя на каждом шаге исполнения алгоритма должны быть достаточно простыми, чтобы их можно было выполнить точно и за конечное время. Обычно отдельные указания исполнителю, содержащиеся в каждом шаге алгоритма, называют *командами*. Таким образом, эффективность алгоритма связана с возможностью выполнения каждой команды за конечное время. Совокупность команд, которые могут быть выполнены конкретным исполнителем, называется *системой команд исполнителя*. Следовательно, алгоритм должен быть сформулирован так, чтобы содержать только те команды, которые входят в систему команд исполнителя. Кроме того, эффективность означает, что алгоритм может быть выполнен не просто за конечное, а за разумно конечное время.

Приведенные выше комментарии поясняют *интуитивное понятие алгоритма*, но само это понятие не становится от этого более четким и строгим. Тем не менее, в математике долгое время использовали это понятие. Лишь с выявлением алгоритмически неразрешимых задач, т.е. задач, для решения которых невозможно построить алгоритм, появилась настоятельная потребность в построении формального определения алгоритма, соответствующего известному интуитивному понятию. Интуитивное понятие алгоритма в силу своей неопределенности не может быть объектом математического изучения, поэтому для доказательства существования или не существования алгоритма решения задачи было необходимо строгое формальное определение алгоритма.

### Основные алгоритмические структуры.

Основными алгоритмическими структурами являются:

- следование;
- ветвление;
- цикл.

«**Следование**» — это часть алгоритма, в которой все команды исполняются одна за другой в порядке их записи.

Линейным называется алгоритм, выполнение шагов которого происходит последовательно в порядке возрастания их номеров. В схеме он изображается последовательностью вычислительных блоков и блоков ввода-вывода.

Конструкция следования:



«**Ветвление**» — это часть алгоритма, в которой выполняется либо одна, либо другая последовательность действий в зависимости от результата проверки условия.

Ветвлением (условием) называется алгоритм, в котором предусмотрено прохождение различных вариантов работы в зависимости от выполнения или не выполнения некоторого условия. В блок-схеме это условие записывается в ромб-блок сравнения.

Различают две формы ветвления:

- полное;
- неполное.



Рис. 2.2. Неполная форма алгоритма ветвления



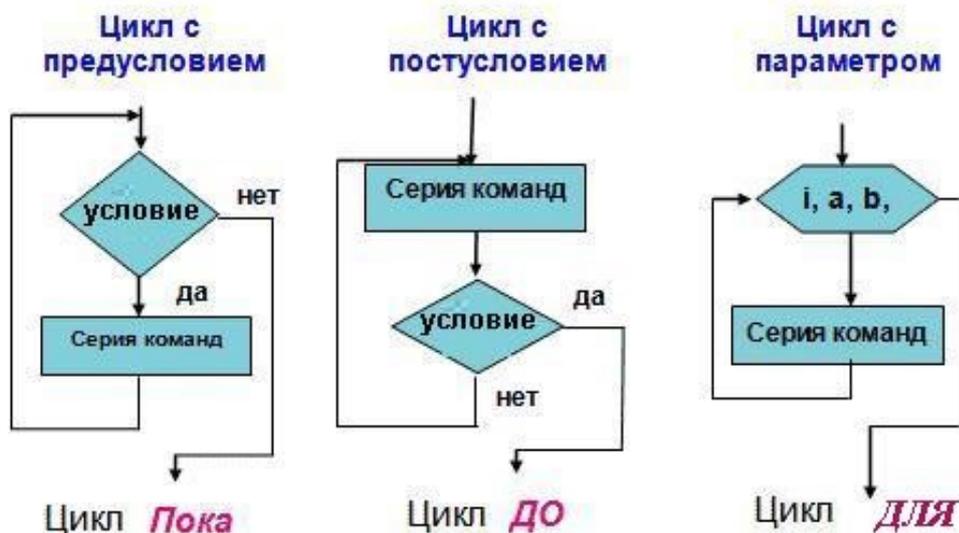
Рис.2.1. Полная форма алгоритма ветвления

«**Цикл**» — это часть алгоритма, в которой некоторую последовательность действий необходимо повторить несколько раз.

Алгоритм циклической структуры — алгоритм, в котором предусмотрено выполнение одной и той же последовательности действий.

Циклом называется участок алгоритма, реализующий многократно повторяющиеся при различных значениях параметров однотипные вычисления (например, расчеты по одной и той же формуле), Алгоритм, содержащий цикл, называется циклическим.

### Виды циклических алгоритмов



Характерной особенностью базовых структур является наличие в них одного входа и одного выхода.

#### Цикл с предусловием (иначе ЦИКЛ ПОКА) имеет вид:

Форматы записи операторов алгоритма	Блок-схема	Форматы записи операторов на Паскале
<p><u>Пока</u> (условие)  НЦ  серия команд  КЦ</p>		<pre>while <i>условие</i> do begin серия команд; end;</pre>

*В цикле «пока» тело цикла выполняется до тех пор, пока выполняется условие.*

**Цикл с постусловием (иначе ЦИКЛ ДО) имеет вид:**

Форматы записи операторов алгоритма	Блок-схема	Форматы записи операторов на Паскале
<p>В алгоритмическом языке нет команды, которая могла бы описать данную структуру, но ее можно выразить с помощью других команд (Например, ветвления).</p>		<p><b>repeat</b> серия команд <b>until</b> условие</p>

В цикле «до» тело цикла выполняется определенное количество раз.

**Цикл с параметром (иначе ЦИКЛ ДЛЯ) имеет вид:**

Форматы записи операторов алгоритма	Блок-схема	Форматы записи операторов на Паскале
<p><u>Для</u> <u>i</u> <u>от</u> <u>a</u> <u>до</u> <u>b</u> <u>шаг</u> <u>h</u> делай <u>Нц</u> Серия команд <u>Кц</u></p>		<p><b><u>h = +1</u></b> <b>for</b> <i>i:= a to b</i> <b>do</b>   <b>begin</b>     серия команд   <b>end;</b> <b><u>h = -1</u></b> <b>for</b> <i>i:= b downto a</i> <b>do</b>   <b>begin</b>     Серия команд;   <b>end;</b></p>

Циклический алгоритм позволяет существенно сократить объем программы.  
Для организации цикла необходимо предусмотреть:

- задание начального значения параметра цикла — переменной, которая будет изменяться при повторениях цикла;
- изменение значения этой переменной перед каждым новым повторением цикла;

- проверку условия окончания повторений по значению параметра и переход к началу цикла, если повторения не закончены.

## Состав средств программирования на языке высокого уровня.

Чаще всего для разработки программ на языках высокого уровня используются интегрированные системы программирования, включающие в себя текстовый редактор, компоненту для перевода исходного текста программы в машинный код, называемую транслятором, и редактор связей.

Каждый язык программирования, как и естественный язык, имеет:

- алфавит – фиксированный для данного языка набор основных символов, допускаемых для составления текста программы на этом языке;
- синтаксис – систему правил, определяющих допустимые конструкции языка программирования из букв алфавита;
- семантику – систему правил однозначного толкования отдельных языковых конструкций, позволяющих воспроизвести процесс обработки данных.

При описании языка и его применении используют понятия языка. Понятие подразумевает некоторую синтаксическую конструкцию и определяемые ею свойства программных объектов или процесса обработки данных.

Взаимодействие синтаксических и семантических правил определяют те или иные понятия языка, например, операторы, идентификаторы, переменные, функции, процедуры, модули и т.д. В отличие от естественных языков правила грамматики и семантики для языков программирования, как и для всех формальных языков, должны быть явно, однозначно и четко сформулированы. Нарушение формы записи программы приводит к синтаксической ошибке, а формально правильно написанная, но не отвечающая алгоритму программа содержит семантическую (логическую) ошибку.

## ТРАНСЛЯЦИЯ, КОМПИЛЯЦИЯ И ИНТЕРПРЕТАЦИЯ

**Транслятор** – это программа или техническое средство, выполняющее преобразование программы, представленной на одном из языков программирования, в программу на другом языке, в определенном смысле равносильную первой.

**Компиляция** в программировании – преобразование программы, представленной на одном из языков программирования, в коды на машинно-

ориентированном языке, которые принимаются и исполняются непосредственно процессором. **Результатом компиляции является объектный файл** с необходимыми внешними ссылками для компоновщика. Программа уже переведена в машинные инструкции, однако еще не полностью готова к выполнению. Даже если в программе явно не упомянута ни одна функция, необходимо, по крайней мере, один вызов системной функции – завершение программы и освобождение всех принадлежащих ей ресурсов.

**Компилятор** – это программа, предназначенная для трансляции исходного текста программы с высокоуровневого языка в объектный код. Входной информацией для компилятора является описание алгоритма или программа на языке программирования. На выходе компилятора – эквивалентное описание алгоритма на машинно-ориентированном языке (объектный код).

**Компоновка** – это один из этапов создания исполняемого файла. Этот этап выполняет:

**Компоновщик** – модуль системы программирования или самостоятельная программа, которая собирает результирующую программу из объектных модулей и стандартных библиотечных модулей. Этот процесс называется компоновкой, его результатом и будет исполняемый файл. С процедурой интерпретации компоновка не связана.

При работе с программами существуют этапы:

- а) компиляции
- б) компоновки
- в) интерпретации
- г) исполнения программы.

### **СОЗДАНИЕ ИСПОЛНЯЕМОГО ФАЙЛА ИЗ ИСХОДНОГО ТЕКСТА ПРОГРАММЫ ПРЕДПОЛАГАЕТ ВЫПОЛНЕНИЕ ЭТАПОВ, А И Б (КОМПИЛЯЦИИ И КОМПОНОВКИ).**

**Исполняемый файл** – это файл, который может быть обработан или выполнен компьютером без предварительной трансляции. Обычно исполняемый файл получается в результате компиляции и компоновки объектных модулей и содержит машинные команды и/или команды операционной системы.

**Интерпретатор** анализирует и тут же выполняет программу покомандно, по мере поступления ее исходного кода на вход интерпретатора.

#### **Алгоритм работы простого интерпретатора:**

- 1) прочитать инструкцию;
- 2) проанализировать инструкцию и определить соответствующие действия;
- 3) выполнить соответствующие действия;
- 4) если не достигнуто условие завершения программы, прочитать следующую инструкцию и перейти к пункту 2.

Утверждение «Языковой процессор, который построчно анализирует исходную программу и одновременно выполняет предписанные действия, а не формирует на машинном языке скомпилированную программу, которая выполняется впоследствии» справедливо для интерпретатора

Режим интерпретации можно использовать при отладке программ на языке высокого уровня.

**Интерпретация в разработке программ** – процесс непосредственного покомандного выполнения программы без предварительной компиляции, «на лету». В большинстве случаев интерпретируемая программа работает намного медленнее, чем скомпилированная программа, но не требует затрат на компиляцию, что в случае небольших программ может повышать общую производительность. Интерпретация связана с получением переменными значений в процессе работы программы.

Интерпретация и компиляция не позволяют изменить семантику (смысл) языка программирования. Обычно **при описании семантики** в рамках операционного подхода исполнение конструкций языка программирования трактуется с помощью некоторой воображаемой (абстрактной) ЭВМ.

Интерпретация и компиляция не позволяют изменить **синтаксис языка** – набор правил построения фраз алгоритмического языка, позволяющий определить осмысленные предложения в этом языке.

**Синтаксический анализ** - это процесс сопоставления линейной последовательности лексем (слов) языка с его формальной грамматикой, обычно это программа или часть программы, выполняющая синтаксический анализ. Результатом обычно является дерево разбора.

**Дисплейный файл** – набор команд дисплейного процессора (представление интерактивной графики). В составе команд дисплейного процессора могут предусматриваться не только графические команды формирования изображений и перехода, но и команды организации циклов и обращения к подпрограммам.

**Командные файлы** используются, когда в процессе работы необходимо часто выполнять одни и те же действия. Операционная система позволяет записать нужную для этого последовательность команд в специальный файл, называемый командным файлом.

## ЭВОЛЮЦИЯ И КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ. ОСНОВНЫЕ ПОНЯТИЯ ЯЗЫКОВ

**Выражение на языке программирования** задает правило вычисления некоторого значения. Выражение **состоит из** констант, переменных, указателей функций, знаков операций и скобок. Каждое выражение **имеет тип**, зависящий от типов, входящих в него операндов.

**Выражение** называют **арифметическим**, если его значением является число. Арифметическое выражение записывается в строку. Нельзя ставить подряд два знака арифметических операций. Нельзя опускать знак умножения.

При вычислении значений выражений важен **порядок выполнения операций**. Приняты следующие правила. Действия выполняются слева направо с соблюдением следующего старшинства (**в порядке убывания приоритетов**):

- 1) действия в скобках;
- 2) вычисление функций;
- 3) NOT;
- 4) \*, /, DIV, MOD, AND;
- 5) +, -, OR;
- 6) =, <>, <, >, <=, >=.

Если приоритет операций одинаков, то операция, стоящая левее, выполняется раньше. Порядок выполнения операций можно переопределить с помощью скобок.

**Язык программирования** - формальная знаковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор **лексических, синтаксических и**

**семантических правил**, задающих внешний вид программы и действия, которые выполнит исполнитель (компьютер) под ее управлением.

**Язык программирования** – искусственный язык, который имеет

- ограниченное число слов, значение которых понятно транслятору (ключевые слова)
- очень строгие правила записи команд

**Синтаксис языка** – совокупность правил записи команд

**Семантика языка** – смысл каждой команды и других конструкций языка

Каждый язык программирования поддерживает один или несколько **встроенных типов данных** (базовых типов). Кроме того, развитые языки программирования предоставляют программисту возможность описывать собственные типы данных, комбинируя или расширяя существующие.

Тип данных – **характеристика набора данных, которая определяет:**

- диапазон возможных значений данных из набора;
- допустимые операции, которые можно выполнять над этими значениями;
- способ хранения этих значений в памяти.

**Различают:**

- -простые типы данных: целые, действительные числа и др.;
- - составные типы данных: массивы, файлы и др.

**Переменная** – именованная область памяти, хранящая некоторое значение.

Массивы используются во всех современных языках программирования.

**Массив** – именованный набор однотипных данных, расположенных в памяти непосредственно друг за другом (в отличие от **списка**), доступ к которым осуществляется по индексу элемента массива (номеру по порядку, упорядоченному по возрастанию в индексных массивах). Индекс массива, указывающий на элемент массива, – это целое число.

**Программа** – логически упорядоченная последовательность команд, необходимых для решения определенной задачи.

**Программа** – алгоритм, записанный на языке программирования.

**Текст программы** – полное, законченное и детальное описание алгоритма на языке программирования. Затем этот текст программы обрабатывается специальными служебными приложениями, которые называются трансляторами.

Команды, поступающие в процессор по его шинам, на самом деле являются электрическими сигналами. Эти сигналы можно интерпретировать как нули и единицы, то есть как числа. Поэтому реально программа, с которой работает процессор, представляет собой последовательность чисел, называемую **машинным кодом**.

**Тестирование программного обеспечения** - процесс исследования программного обеспечения (ПО) с целью получения информации о качестве продукта (в том числе поиск ошибок в программе).

**Альфа-тестирование** – тестирование готового продукта на специально созданных задачах.

**Бета-тестирование** – опробование бесплатной тестовой версии программного продукта на реальных задачах.

Если «**альфа-»** и «**бета-тестирование**» относятся к стадиям до **выпуска продукта** (а также, неявно, к объёму тестирующего сообщества и ограничениям на методы тестирования), **тестирование «белого ящика»** и «**чёрного ящика»** имеет отношение к способам, которыми **тестировщик** достигает цели.

При **тестировании белого ящика** (англ. *white-box testing*, также говорят - *прозрачного ящика*), разработчик теста имеет доступ к исходному коду программ и может писать код, который связан с библиотеками тестируемого ПО. Это типично для юнит-тестирования (англ. *unit testing*), при котором тестируются только отдельные части системы. Оно обеспечивает то, что компоненты конструкции — работоспособны и устойчивы, до определённой степени. При тестировании белого ящика используются метрики покрытия кода.

При **тестировании чёрного ящика**, тестировщик имеет доступ к ПО только через те же интерфейсы, что и заказчик или пользователь, либо через внешние интерфейсы, позволяющие другому компьютеру либо другому процессу подключиться к системе для тестирования. Например, тестирующий модуль может виртуально нажимать клавиши или кнопки мыши в тестируемой программе с помощью механизма взаимодействия процессов, с уверенностью в том, все ли идёт правильно, что эти события вызывают тот же отклик, что и реальные нажатия клавиш и кнопок мыши. Как правило, тестирование чёрного ящика ведётся с использованием спецификаций или иных документов, описывающих требования к системе. Как правило, в данном виде тестирования критерий покрытия складывается

из покрытия структуры входных данных, покрытия требований и покрытия модели (в тестировании на основе моделей).

**Регрессионное тестирование.** После внесения изменений в очередную версию программы, регрессионные тесты подтверждают, что сделанные изменения не повлияли на работоспособность остальной функциональности приложения. Регрессионное тестирование может выполняться как вручную, так и средствами автоматизации тестирования.

**Отладка** - этап разработки компьютерной программы, на котором обнаруживают, локализуют и устраняют ошибки. Чтобы понять, где возникла ошибка, приходится:

- узнавать текущие значения переменных;
- и выяснять, по какому пути выполнялась программа.

Существуют две **взаимодополняющие технологии отладки**.

- **Использование отладчиков-программ**, которые включают в себя пользовательский интерфейс для пошагового выполнения программы: оператор за оператором, функция за функцией, с остановками на некоторых строках исходного кода или при достижении определённого условия.
- **Вывод текущего состояния программы** с помощью расположенных в критических точках программы операторов вывода — на экран, принтер, громкоговоритель или в файл. Вывод отладочных сведений в файл называется **журналированием**.

**Отладчик** — программа, позволяющая исследовать внутреннее устройство программы. Отладчик обеспечивает пошаговое исполнение программы, просмотр текущих значений переменных, вычисление значения любого выражения программы и др. **Debugger**.

**Рефакторинг** - процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и имеющий целью облегчить понимание её работы

**ПОДПРОГРАММА** — набор операторов, выполняющих нужное действие и не зависящих от других частей исходного кода.

**Подпрограммы** бывают двух видов – **процедуры и функции**. Они отличаются тем, что процедура просто выполняет группу операторов, а функция к тому же передает в главную программу некоторое вычисленное значение. Передает = возвращает. Функция имеет тип, совпадающий с типом возвращаемого значения.

Внешняя программа вызывает подпрограмму. Данные передаются подпрограмме в виде параметров или аргументов, которые описываются в заголовке подпрограммы.

Подпрограммы могут быть вложенными, т.е. допускается вызов подпрограммы не только из главной программы, но и из другой

подпрограммы. Подпрограмма может вызывать сама себя. Такой прием называется **рекурсией**.

Важная характеристика подпрограмм – возможность их повторного использования. С интегрированными системами программирования поставляются большие библиотеки стандартных подпрограмм.

**Модуль** – процедура, функция, юнит, программа в целом – логически законченная составная часть программы.

### Инструментальные системы программирования

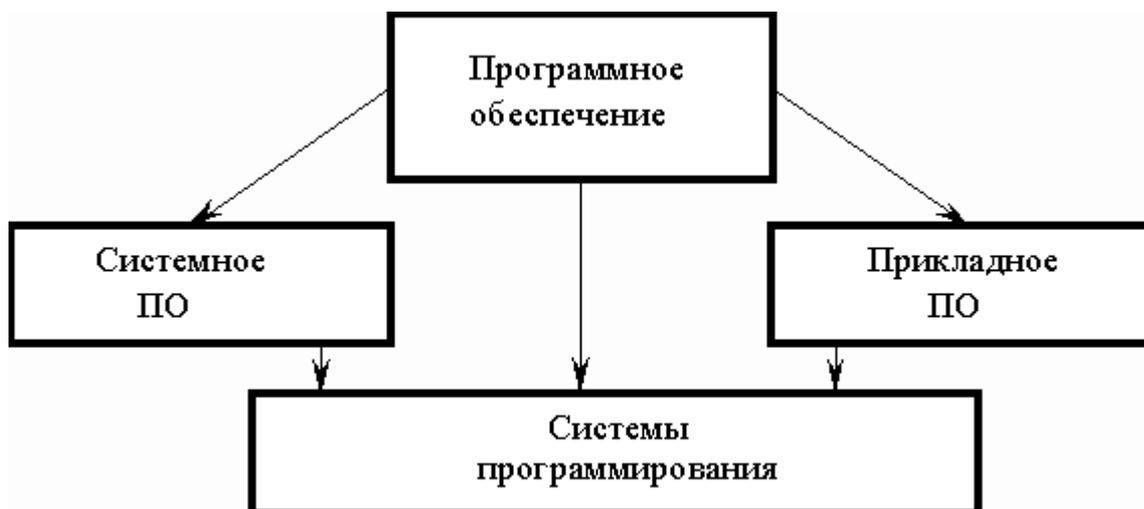
Для популярных языков программирования на ЭВМ существует множество систем программирования. Программисты предпочитают те системы, которые легки в использовании, позволяют получить эффективные программы, имеют богатые библиотеки функций (подпрограмм) и мощные возможности для отладки разрабатываемых программ. В качестве примеров таких систем программирования можно назвать **Delphi, Visual C++, Visual Basic**.

Системы программирования прежде всего различаются по тому, какой язык программирования они реализуют. Среди программистов, пишущих программы для персональных компьютеров, наибольшей популярностью пользуются языки **Си, Паскаль и Бейсик**.

### Классификация программного обеспечения

Программное обеспечение (ПО) - это совокупность всех программ и соответствующей документации, обеспечивающая использование ЭВМ в интересах каждого ее пользователя.

Различают системное и прикладное ПО. Схематически программное обеспечение можно представить так:



*Системное ПО* – это совокупность программ для обеспечения работы компьютера. Системное ПО подразделяется на базовое и сервисное. Системные программы предназначены для управления работой вычислительной системы, выполняют различные вспомогательные функции (копирования, выдачи справок, тестирования, форматирования и т.д.).

Базовое ПО включает в себя:

- операционные системы;
- оболочки;
- сетевые операционные системы.

Сервисное ПО включает в себя программы (утилиты):

- диагностики;
- антивирусные;
- обслуживания носителей;
- архивирования;
- обслуживания сети.

*Прикладное ПО* – это комплекс программ для решения задач определённого класса конкретной предметной области. Прикладное ПО работает только при наличии системного ПО.

Прикладные программы называют приложениями. Они включает в себя:

- текстовые процессоры;
- табличные процессоры;
- базы данных;
- интегрированные пакеты;
- системы иллюстративной и деловой графики (графические процессоры);
- экспертные системы;
- обучающие программы;

- программы математических расчетов, моделирования и анализа;
- игры;
- коммуникационные программы.

Особую группу составляют системы программирования (инструментальные системы), которые являются частью системного ПО, но носят прикладной характер. Системы программирования – это совокупность программ для разработки, отладки и внедрения новых программных продуктов. Системы программирования обычно содержат:

- трансляторы;
- среду разработки программ;
- библиотеки справочных программ (функций, процедур);
- отладчики;

### Краткий обзор прикладного программного обеспечения

Рассмотрим наиболее часто встречающееся прикладное ПО.

Редакторы документов – это наиболее широко используемый вид прикладных программ. Они позволяют подготавливать документы гораздо быстрее и удобнее, чем с помощью пишущей машинки. Редакторы документов позволяют использовать различные шрифты символов, абзацы произвольной формы, автоматически переносят слова на новую строку, позволяют делать сноски, включать рисунки, автоматически нумеруют страницы и сноски и т.д. Представители редакторов документов – программы **Microsoft Word**, **Wordpad**.

Табличные процессоры. При работе с табличным процессором на экран выводится прямоугольная таблица, в клетках которой могут находиться числа, пояснительные тексты и формулы для расчета значения в клетке по именуемому данным. Все распространенные табличные процессоры позволяют вычислять значения элементов таблиц по заданным формулам, строить по данным в таблицах различные графики и т.д. Представители семейства табличных процессоров **Microsoft Excel**, **Quatro Pro**.

Графические редакторы позволяют создавать и редактировать рисунки. В простейших редакторах предоставляются возможности рисования линий, кривых, раскраски областей экрана, создание надписей различными шрифтами и т.д. Большинство редакторов позволяют обрабатывать изображения, полученные с помощью сканеров. Представители графических редакторов – программы **Adobe Photoshop**, **Corel Draw**.

Правовые базы данных содержат тексты нормативных документов и предоставляют возможности справки, контекстного поиска, распечатки и т.д. Представители правовых баз данных – пакеты Гарант и Консультант+ .

Системы автоматизированного проектирования (САПР) позволяют осуществлять черчение и конструирование различных предметов и механизмов с помощью компьютера. Среди систем малого и среднего класса в мире наиболее популярна система **AutoCad** фирмы **AutoDesk**. Отечественный пакет с аналогичными функциями – **Компас**.

Системы управления базами данных (СУБД) позволяют управлять большими информационными массивами - базами данных. Программные системы этого вида позволяют обрабатывать на компьютере массивы информации, обеспечивают ввод, поиск, сортировку выборку записей, составление отчетов и т.д. Представители данного класса программ – **Microsoft Access, Clipper, Paradox**.

Интегрированные системы сочетают в себе возможность системы управления базами данных, табличного процессора, текстового редактора, системы деловой графики, а иногда и другие возможности. Как правило, все компоненты интегрированной системы имеют схожий интерфейс, что облегчает обучение работе с ними. Представители интегрированных систем – пакет **Microsoft Office** и его бесплатный аналог **Open Office**.

Бухгалтерские программы предназначены для ведения бухгалтерского учета, подготовки финансовой отчетности и финансового анализа деятельности предприятий. Из-за несовместимости отечественного бухгалтерского учета с зарубежным в нашей стране используются почти исключительно отечественные бухгалтерские программы. Наиболее распространены системы **1С: Предприятие** и **Инфо-бухгалтер**.

### Вспомогательные программы

Вспомогательные программы (утилиты) обычно предназначены не для решения конкретных пользовательских задач, а для обслуживания и повышения эффективности вычислительной системы. Кратко остановимся на основных видах вспомогательных программ.

Программы-архиваторы позволяют за счет применения специальных алгоритмов упаковки информации сжимать информацию на дисках, т.е. создавать копии файлов меньшего размера, а также объединять копии нескольких файлов в один архивный файл. Применение программ-архиваторов очень полезно при создании архива файлов, так как в большинстве случаев значительно удобнее их хранить, предварительно сжав

программами-архиваторами. Представители данных программ – WinRar и WinZip.

Программы для создания резервных копий информации позволяют периодически копировать важную информацию, находящуюся на жестком диске компьютера, на дополнительные носители. Представители программ резервного копирования – APBackUp, Acronis True Image.

Антивирусные программы предназначены для предотвращения заражения компьютерными вирусами и ликвидации последствий заражения вирусом. Представители антивирусного семейства программ – Kaspersky Antivirus, DrWeb, Norton Antivirus.

Коммуникационные программы предназначены для организации обмена информацией между компьютерами. Это программы позволяют удобно пересылать файлы с одного компьютера на другой при соединении кабелем их последовательных портов. Другой вид таких программ обеспечивает возможность связи компьютеров по телефонной сети (при наличии модема). Они дают возможность посылать и принимать телефаксные сообщения. Представители коммуникационных программ – Venta Fax, Cute FTP.

Программы для диагностики компьютера позволяют проверить конфигурацию компьютера (количество памяти, ее использование, типы дисков и т. д.), проверить работоспособность устройств компьютера, оценить его производительность. Представители программ диагностики компьютеров – Sisoft Sandra, Norton System Information.

Программы для оптимизации дисков позволяют обеспечить более быстрый доступ к информации на диске за счет оптимизации размещения данных на диске. Эти программы перемещают все участки каждого файла друг к другу (устраняют фрагментацию), собирают все файлы в начале диска и т.д., за счет чего уменьшается число перемещений головок диска (т.е. ускоряется доступ к данным) и снижается износ диска. Представители программ для оптимизации дисков - Norton Disk Doctor, Microsoft Scandisk.

Программы для печати экрана бывают весьма полезны при использовании графических программ для вывода на печать содержимого экрана, так как отнюдь не всегда это можно сделать с помощью самой графической программы. Представители программ для печати экрана – SnagIt, HyperSnap-DX.

## Операционные системы

Операционная система – это комплекс программ, обеспечивающих управление работой компьютера и его взаимодействие с пользователем.

С точки зрения человека операционная система служит посредником между человеком, электронными компонентами компьютера и прикладными программами. Она позволяет человеку запускать программы, передавать им и получать от них всевозможные данные, управлять работой программ, изменять параметры компьютера и подсоединённых к нему устройств, перераспределять ресурсы. Работа на компьютере фактически является работой с его операционной системой. При установке на компьютер только операционной системы (ОС) ничего содержательного на компьютере также сделать не удастся. Для ввода и оформления текстов, рисования графиков, расчёта зарплаты или прослушивания лазерного диска нужны специальные прикладные программы. Но и без ОС ни одну прикладную программу запустить невозможно.

Операционная система решает задачи, которые можно условно разделить на две категории:

- во-первых, управление всеми ресурсами компьютера;
- во-вторых, обмен данными между устройствами компьютера, между компьютером и человеком.

Кроме того, именно ОС обеспечивает возможность индивидуальной настройки компьютера: ОС определяет, из каких компонентов собран компьютер, на котором она установлена, и настраивает сама себя для работы именно с этими компонентами.

Ещё не так давно работы по настройке приходилось выполнять пользователю вручную, а сегодня производители компонентов компьютерной техники разработали протокол plug-and-play (включил - заработало). Этот протокол позволяет операционной системе в момент подключения нового компонента получить информацию о новом устройстве, достаточную для настройки ОС на работу с ним.

Операционные системы для ПК различаются по нескольким параметрам. В частности, ОС бывают:

- однозадачные и многозадачные;
- однопользовательские и многопользовательские;
- сетевые и несетевые.

Кроме того, операционная система может иметь командный или графический многооконный интерфейс (или оба сразу).

Однозадачные операционные системы позволяют в каждый момент времени решать только одну задачу. Такие системы обычно позволяют запустить одну программу в основном режиме.

Многозадачные системы позволяют запустить одновременно несколько программ, которые будут работать параллельно.

Главным отличием многопользовательских систем от однопользовательских является наличие средств защиты информации каждого пользователя от несанкционированного доступа других пользователей. Следует заметить, что не всякая многозадачная система является многопользовательской, и не всякая однопользовательская ОС является однозадачной.

В последние годы фактическим стандартом стал графический многооконный интерфейс, где требуемые действия и описания объектов не вводятся в виде текста, а выбираются из меню, списков файлов и т.д.

В настоящее время, с появлением мощных компьютеров, широкое распространение получили два типа ОС. К первому типу относятся достаточно похожие ОС семейства Windows компании Microsoft. Они многозадачные и имеют многооконный графический интерфейс. На рынке персональных компьютеров с Windows конкурируют ОС типа UNIX. Это многозадачная многопользовательская ОС с командным интерфейсом. В настоящее время разработаны расширения UNIX, обеспечивающие многооконный графический интерфейс. UNIX развивалась в течение многих лет разными компаниями, но до недавнего времени она не использовалась на персональных компьютерах, т.к. требует очень мощного процессора, весьма дорога и сложна, её установка и эксплуатация требуют высокой квалификации. В последние годы ситуация изменилась. Компьютеры стали достаточно мощными, появилась некоммерческая, бесплатная версия системы UNIX для персональных компьютеров - система Linux. По мере роста популярности этой системы в ней появились дополнительные компоненты, облегчающие её установку и эксплуатацию. Немалую роль в росте популярности Linux сыграла мировая компьютерная сеть Internet. Хотя освоение Linux гораздо сложнее освоения систем типа Windows. Linux - более гибкая и в то же время бесплатная система, что и привлекает к ней многих пользователей.

Существуют и другие ОС. Известная компания Apple производит компьютеры Macintosh с современной ОС MacOS. Эти компьютеры используются преимущественно издателями и художниками. Фирма IBM производит ОС OS/2. Операционная система OS/2 такого же класса надёжности и защиты, как и Windows NT.

## Операционная система Windows

На смену операционной системе MS DOS с ее графическими оболочками Windows 3.1 и Windows 3.11 пришли полноценные операционные системы семейства Microsoft Windows (сначала Windows 95, затем Windows 98, Windows Millennium, Windows 2000, Windows XP, Windows Vista. Windows 7, Windows Server 2008, Windows 8 и Windows 8.1).

**Windows 10** - операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT. После Windows 8.1 система получила номер 10, минуя 9. Серверный аналог Windows 10 - Window Server 2016.

Система призвана стать единой для разных устройств, таких как персональные компьютеры, планшеты, смартфоны, консоли Xbox One и пр. Доступна единая платформа разработки и единый магазин универсальных приложений, совместимых со всеми поддерживаемыми устройствами. Windows 10 поставляется в качестве услуги с выпуском обновлений на протяжении всего цикла поддержки. В течение первого года после выхода системы пользователи могли бесплатно обновиться до Windows 10 на устройствах под управлением лицензионных копий Windows 7, Windows 8.1 и Windows Phone 8.1. Среди значимых нововведений - голосовая помощница Кортана, возможность создания и переключения нескольких рабочих столов и др. Windows 10 - последняя «коробочная» версия Windows, все последующие версии будут распространяться исключительно в цифровом виде.

Пользовательское соглашение Windows 10 позволяет компании Microsoft собирать многочисленные сведения о пользователе, историю его интернет-деятельности, пароли к точкам доступа, данные, набираемые на клавиатуре, и многое другое.

**Операционная среда Windows** разработана фирмой Microsoft для IBM-совместимых компьютеров.

**Windows** выполняет следующие основные функции:

- Удобный, наглядный графический интерфейс пользователя.
- Многозадачная работа, т.е. выполнение одновременно нескольких программ.
- Унификация использования аппаратных ресурсов компьютера.

**Особенности Windows:**

- **Правила написания программы.** Для работы в среде Windows программа должна быть написана по определенным правилам, существенно отличающимся от принятых в MS-DOS.

Windows позволяет запускать и программы, написанные для MS DOS, но при этом программы не могут использовать преимущества Windows.

- **Графический интерфейс** пользователя в Windows основан на идее оконного интерфейса, принятого так же и в ряде других современных ОС (например, UNIX). Каждая программа имеет собственное окно, в котором и происходит обмен сообщений с пользователем. Для наглядности в Windows широко применяются иконки (пиктограммы), изображающие отдельные программы.

Кроме того, интерфейс Windows в значительной степени стандартизирован, что облегчает пользователям процесс освоения новых программ.

- **Многозадачность.** Многозадачный режим работы позволяет запускать одновременно несколько приложений, например, текстовый процессор, базу данных, игру и переключаться между ними.
- **Обмен данными между приложениями.** Кроме того возможен обмен данными между приложениями, что позволяет, например, информацию созданную в электронной таблице, перенести в текстовый документ через буфер обмена.

Операционные системы семейства Windows представляет собой 32-разрядные операционные системы, обеспечивающую многозадачную и многопоточную обработку приложений. Они поддерживает удобный графический пользовательский интерфейс, возможность работы в защищенном режиме, совместимость с программами реального режима и сетевые возможности. В Windows реализована технология поддержки самонастраивающейся аппаратуры Plug and Play, допускаются длинные имена файлов и обеспечиваются повышенные характеристики устойчивости.

32-разрядность означает, что операции над 32-разрядными данными здесь выполняются быстрее, чем над 16-разрядными. 32-разрядные Windows-приложения выполняются в собственном адресном пространстве, доступ в которое для других программ закрыт. Это защищает приложения от ошибок друг друга. При сбое в работе одного приложения другое продолжает нормально функционировать. Сбойное же приложение можно завершить.

Многозадачность предоставляет возможность параллельной работы с несколькими приложениями. Пока одно из них занимается, например, печатью документа на принтере или приемом электронной почты из сети Internet,

другое может пересчитывать электронную таблицу или выполнять другую полезную работу.

Многопоточность позволяет определенным образом разработанным приложениям одновременно выполнять несколько своих собственных процессов. Например, работая с многопоточной электронной таблицей, пользователь сможет делать перерасчет в одной таблице в то время, как будет выполняться печать другой и загрузка в память третьей. Пока один поток находится в состоянии ожидания, например, завершения операции обмена данными с медленным периферийным устройством, другой может продолжать выполнять свою работу.

Отличительной чертой Windows является объектно-ориентированный подход к построению системы. На уровне пользователя объектный подход выражается в том, что интерфейс представляет собой подобие реального мира, а работа с машиной сводится к действиям с привычными объектами. Так, папки можно открыть, убрать в портфель, документы – просмотреть, исправить, переложить с одного места на другое, выбросить в корзину, факс или письмо – отправить адресату и т. д. Пользователь работает с задачами и приложениями так же, как с документами на своем письменном столе. Объектно-ориентированный подход реализуется через модель рабочего стола – первичного объекта Windows. После загрузки Windows он выводится на экран. На рабочем столе могут быть расположены различные объекты: программы, папки с документами (текстами, рисунками, таблицами), ярлыки программ или папок.

Ярлыки обеспечивают доступ к программе или документу из различных мест, не создавая при этом нескольких физических копий файла. На рабочий стол можно поместить не только пиктограммы приложений и отдельных документов, но и папок. Папки - еще одно название каталогов.

Существенным нововведением в Windows стала панель задач. Несмотря на небольшие функциональные возможности, она делает наглядным механизм многозадачности и намного ускоряет процесс переключения между приложениями. Внешне панель задач представляет собой полосу, обычно располагающуюся в нижней части экрана, на которой размещены кнопки приложений и кнопка “Пуск”. В правой ее части обычно присутствуют часы и небольшие пиктограммы программ, активных в данный момент.

Windows обеспечивает работу с аудио и видеофайлами различных форматов. Значительным достижением Windows стали встроенные в систему программы для компьютерных коммуникаций. Коммуникационные средства Windows рассчитаны на обычных пользователей и не требуют специальных знаний. Эти средства включают в себя возможности работы в локальных сетях

и глобальных сетях, настройку модемов, подключение к электронной почте и многое другое.

В операционной системе Windows при работе с окнами и приложениями широко применяется манипулятор мышь. Обычно мышь используется для выделения фрагментов текста или графических объектов, установки и снятия флажков, выбора команд меню, кнопок панелей инструментов, манипулирования элементами управления в диалогах, "прокручивания" документов в окнах.

В Windows активно используется и правая кнопка мыши. Поместив указатель над интересующем объекте и сделав щелчок правой кнопкой мыши, можно раскрыть контекстное меню, содержащее наиболее употребительные команды, применимые к данному объекту.

При завершении работы нельзя просто выключить компьютер, не завершив работу системы по всем правилам - это может привести к потере некоторых несохраненных данных. Для правильного завершения работы необходимо сохранить данные во всех приложениях, с которыми работал пользователь, завершить работу всех ранее запущенных DOS-приложений, открыть меню кнопки "Пуск" и выбрать команду "Завершение работы".

## 10.1. Информационно-поисковые системы и их классификация

Информационно-поисковая система – это прикладная компьютерная среда для обработки, хранения, сортировки, фильтрации и поиска больших массивов структурированной информации.

Каждая ИПС предназначена для решения определенного класса задач, для которых характерен свой набор объектов и их признаков. ИПС бывают двух типов:

1. Документографические. В документографических ИПС все хранимые документы индексируются специальным образом, т. е. каждому документу присваивается индивидуальный код, составляющий поисковый образ. Поиск идет не по самим документам, а по их поисковым образам. Именно так ищут книги в больших библиотеках. Сначала отыскивают карточку в каталоге, а затем по номеру, указанному на ней, отыскивается и сама книга.

2. Фактографические. В фактографических ИПС хранятся не документы, а факты, относящиеся к какой-либо предметной области. Поиск осуществляется по образцу факта.

Каждая ИПС состоит из двух частей: базы данных (БД) и системы управления базами данных (СУБД).

База данных - это поименованная совокупность структурированных данных, относящихся к определенной предметной области.

Система управления базами данных - это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации.

На настоящий момент существует множество различных СУБД. Наиболее широкую известность получили такие как Dbase, Clipper, FoxPro, Paradox, Microsoft Access.

## 10.2. Информационные единицы баз данных

Объектами обработки СУБД являются следующие информационные единицы.

Поле - элементарная единица логической организации данных, которая соответствует неделимой единице информации - реквизиту.

Запись - совокупность логически связанных полей.

Экземпляр записи - отдельная реализация записи, содержащая конкретные значения ее полей.

Таблица - упорядоченная структура, состоящая из конечного набора однотипных записей.

Первичный ключ - поле или группа полей, позволяющие однозначным образом определить каждую строку в таблице. Первичный ключ должен обладать двумя свойствами:

1. Однозначная идентификация записи: запись должна однозначно определяться значением ключа.
2. Отсутствие избыточности: никакое поле нельзя удалить из ключа, не нарушая при этом свойства однозначной идентификации.

Кроме первичного, могут использоваться так называемые простые (или вторичные) ключи таблицы. Простых ключей может быть множество. Они используются при упорядочивании (индексировании) таблиц.

## Эволюция и классификация языков программирования

Таблица 4 Эволюция ЯП

Поколения	Языки программирования	Характеристика
Первое	Машинные	Ориентированы на использование в конкретной ЭВМ, сложны в освоении, требуют хорошего знания архитектуры ЭВМ
Второе	Ассемблеры, макроассемблеры	Более удобны для использования, но по-прежнему машинно-зависимы
Третье	Языки высокого уровня	Мобильные, человеко-ориентированные, проще в освоении
Четвёртое	Непроцедурные, объектно-ориентированные, языки запросов, параллельные	Ориентированы на непрофессионального пользователя и на ЭВМ с параллельной архитектурой
Пятое	Языки искусственного интеллекта, экспертных систем и баз знаний, естественные языки	Ориентированы на повышение интеллектуального уровня ЭВМ и интерфейса с языками

Таблица 5. Классификация ЯП

Фактор	Характеристика	Группы	Примеры ЯП
Уровень ЯП	Степень близости ЯП к архитектуре ПК	Низкий	Автокод, ассемблер
		Высокий	Fortran, Pascal, ADA, Basic, C и др.
		Сверхвысокий	Сетл
Специализация ЯП	Потенциальная или реальная область применения	Общего назначения (универсальные)	Fortran (инженерные расчёты), Cobol (Коммерческие задачи), Refal, Lisp(символьная обработка), Modula, ADA(программирование в реальном времени)
		Специализированные	
Алгоритмичность (процедурность)	Возможность абстрагироваться от деталей алгоритма решения задачи. Алгоритмичность тем выше, чем точнее приходится планировать порядок выполняемых действий	Процедурные	Ассемблер, Fortran, Basic, Pascal, ADA
		Непроцедурные	Prolog, Langin

### Структурное программирование.

*Структурное программирование* — это метод программирования, опирающийся на структурную организацию программы.

**Основной принцип** структурного программирования - обеспечить максимальное соответствие структуры текста программы логике решаемой проблемы.

Основным принципом технологии структурного программирования является *нисходящее программирование* - это программирование с использованием подпрограмм, которое позволяет вести разработку приложения "сверху вниз", от более общих задач к более частным. При проектировании *сверху вниз* программа строится с помощью последовательных уточнений. Этот процесс начинается с самого общего утверждения об ее абстрактной функции, такого как "Оттранслировать программу в машинный код" или "Обработать команду пользователя" и продолжается путем последовательных шагов уточнения. На каждом шаге уровень абстракции получаемых элементов должен уменьшаться, каждая операция на нем разлагается на композицию одной или нескольких более простых операций.

Программирования сверху – вниз.

Сутью структурного программирования является возможность разбиения программы на составляющие элементы.

Распространены две методики (стратегии) разработки программ, относящиеся к структурному программированию: программирование "сверху вниз" и программирование "снизу вверх".

**Программирование "сверху вниз", или нисходящее программирование** – это методика разработки программ, при которой разработка начинается с определения целей решения проблемы, после чего идет последовательная детализация, заканчивающаяся детальной программой. Является противоположной методике программирования «снизу вверх».

При нисходящем проектировании задача анализируется с целью определения возможности разбиения ее на ряд подзадач. Затем каждая из полученных подзадач также анализируется для возможного разбиения на подзадачи. Процесс заканчивается, когда подзадачу невозможно или нецелесообразно далее разбивать на подзадачи.

В данном случае программа конструируется иерархически - сверху вниз: от главной программы к подпрограммам самого нижнего уровня, причем на каждом уровне используются только простые последовательности инструкций, циклы и условные разветвления.

Программирования снизу – вверх.

**Программирование "снизу-вверх", или восходящее программирование** – это методика разработки программ, начинающаяся с разработки подпрограмм (процедур, функций), в то время, когда проработка общей схемы не закончилась. Является противоположной методике программирования «сверху вниз».

Такая методика является менее предпочтительной по сравнению с нисходящим программированием так как часто приводит к нежелательным результатам, переделкам и увеличению времени разработки.

Способы реализации основного принципа структурного программирования:

- 1) **Структурируемость и читаемость текста программы** – отступы, обозначения, группировка частей текста.
- 2) **Отлаживаемость программы** – пригодность конструкций языка к отладке. В значительной степени связана с особенностями типизации данных. Наличие жесткого контроля типов с диагностикой ошибок при трансляции программы. Другая важная часть – наличие интегрированной среды разработки.
- 3) **Инкапсуляция данных** — наличие структурных типов данных (массивы, записи, строки, множества, файловые типы, потоки и т.п.) с возможностью оперирования переменной структурного типа как единым целым.
- 4) **Инкапсуляция программного кода** - блоки разных уровней, модули, пакеты и т.п. с особыми правилами прозрачности и интерфейсами между ними.
- 5) **Инкапсуляция программного кода и данных во время выполнения программы**, со специальными ограничениями доступа, правилами прозрачности, интерфейсами, диагностикой ошибок во время выполнения программы. Примеры: обработка исключительных ситуаций во время выполнения, обработки событий, подпроцессы.

Достоинства структурного программирования:

- 1) повышается надежность программ (благодаря хорошему структурированию при проектировании, программа легко поддается тестированию и не создает проблем при отладке);
  - 2) повышается эффективность программ (структурирование программы позволяет легко находить и корректировать ошибки, а отдельные подпрограммы можно переделывать (модифицировать) независимо от других);
  - 3) уменьшается время и стоимость программной разработки;
  - 4) улучшается читабельность программ.
- Структурное программирование эффективно используется для решения различных математических задач, имеющих алгоритмический характер.

## Модульный принцип программирования

**Модульное программирование** — это организация программы как совокупности небольших независимых блоков (модулей), структура и поведение которых подчиняется определенным заранее правилам.

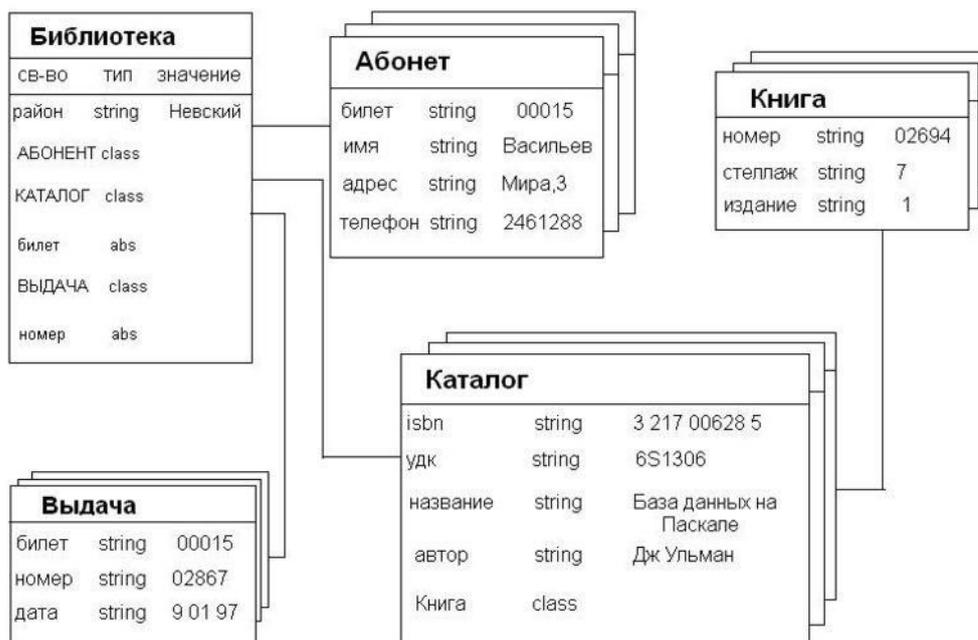
**Модулем** (в модульном программировании) называется множество взаимосвязанных подпрограмм (процедур) вместе с данными, которые эти подпрограммы обрабатывают.

Модульное программирование предназначено для разработки больших программ.

Разработкой больших программ занимается коллектив программистов. Каждому программисту поручается разработка некоторой самостоятельной части программы. И он в таком случае отвечает за конструирование всех необходимых процедур и данных для этих процедур. Скрытие данных (запрет доступа к данным из-за пределов модуля) предотвращает их случайное изменение и соответственно нарушение работы программы. Для взаимодействия отдельных частей (модулей) программы коллективу программистов необходимо продумать только интерфейс (взаимодействие) сконструированных модулей в основной программе.

## Объектно-ориентированное программирование.

В объектно-ориентированной модели данных любая сущность реального мира представляется всего одним понятием — **объектом**. С объектом ассоциируется состояние и поведение. **Состояние объекта** определяется значениями его **свойств** — **атрибутов**.



**Объектно-ориентированное программирование (ООП)**- методология программирования, основанная на представлении программы в виде совокупности **объектов**, каждый из которых является экземпляром определенного **класса**, а классы образуют иерархию наследования

Вместо поиска самой верхней функции системы будут анализироваться типы входящих в нее объектов. Проектирование системы будет продвигаться вперед путем последовательного улучшения понимания классов этих объектов. Это процесс построения снизу-вверх устойчивых и расширяемых решений для отдельных частей задачи и сборки из них все более и более мощных блоков будет продолжаться до тех пор, пока не будет получен окончательный блок, доставляющий решение первоначальной задачи. При этом можно надеяться, что оно не является единственно возможным: если правильно применять метод, то те же компоненты, собранные по-другому и, возможно, объединенные с другими, окажутся достаточно общими, чтобы получить в качестве побочного продукта также и решения каких-то новых задач.

Основные принципы структурирования в случае ООП связаны с различными аспектами базового понимания предметной задачи, которое требуется для оптимального управления соответствующей моделью:

- абстрагирование для выделения в моделируемом предмете важного для решения конкретной задачи по предмету, в конечном счете — контекстное понимание предмета, формализуемое в виде класса;

- инкапсуляция для быстрой и безопасной организации собственно иерархической управляемости: чтобы было достаточно простой команды «что делать», без одновременного уточнения как именно делать, так как это уже другой уровень управления;

- наследование для быстрой и безопасной организации родственных понятий: чтобы было достаточно на каждом иерархическом шаге учитывать только изменения, не дублируя все остальное, учтенное на предыдущих шагах;

- полиморфизм для определения точки, в которой единое управление лучше распараллелить или наоборот — собрать воедино.

Объектно-ориентированное программирование позволяет разложить проблему на составные части. Каждая составляющая становится самостоятельным объектом, содержащим свои собственные коды и данные, которые относятся к этому объекту. В этом случае программирование в целом упрощается, и программист получает возможность оперировать гораздо большими по объёму программами.

Основным понятием ООП является понятие класса.

Класс – множество объектов, связанных общностью структуры и поведения (класс содержит описание структуры и поведение всех объектов,

связанных отношением общности). Любой объект является экземпляром класса.

Методом называется процедура или функция, определенная внутри класса.

Базовые принципы ООП

1. Инкапсуляция
2. Наследование
3. Полиморфизм

### **Инкапсуляция**

Инкапсуляция представляет собой комбинирование данных (записи, структуры) с процедурами и функциями для получения нового типа данных.

Здесь проводится аналогия с физическими объектами. Конкретные физические свойства определяются данными различных типов. Кроме того, любой физический объект характеризуется и своим поведением во внешнем мире. Поведение объекта задается процедурами и функциями.

Итак, инкапсуляция означает, что методы (коды) и данные одновременно представлены в одной и той же структуре.

Например,

Type

```
Coordinates = class  
  x, y : byte;  
  procedure Init (Xinit, Yinit: byte);  
  function GetX : byte;  
  function GetY : byte;  
end;
```

### **Наследование**

Наследование служит для использования однажды определенного класса в построении целой иерархии производных классов, каждый из которых наследует доступ к данным и методам работы с ними всех своих «родителей».

То есть, можно построить иерархию классов, которая выражает родословное дерево классов. Классы организованы в единую древовидную структуру с общим корнем. Свойства и методы определенного класса автоматически доступны любому классу, расположенному ниже в иерархическом дереве.

Таким образом, наследование – это особенность ООП, посредством которой класс может быть определен как расширение уже существующего класса.

Например,

Type

```
Cursor = class (coordinates)
    Hidden : Boolean;
end;
```

Здесь класс «курсор» наследует все свойства и методы предварительно описанного класса «координаты», т.е. для него определены координаты x, y, а также методы «инициализация», «определить координату x», «определить координату y» (см. выше описание класса «координаты»). Кроме того, класс «курсор» обладает собственным методом - «быть невидимым».

Наследование позволяет различным типам данных совместно использовать один и тот же код, приводя к уменьшению его размера и повышению функциональности.

### Полиморфизм

Полиморфизм – это придание действию (методу) одного имени, которое совместно используется объектами всей иерархии класса, причем каждый объект реализует это действие своим собственным, подходящим для него образом.

Другими словами, полиморфизм – это использование одинаковых имен методов на разных уровнях иерархии классов.

### Достоинства ООП

С помощью уменьшения взаимозависимости между компонентами программного обеспечения ООП позволяет разрабатывать системы, пригодные для многократного использования. Такие компоненты могут быть созданы и отлажены как независимые программные единицы.

*Несколько слов об использовании технологии ООП в среде Delphi на языке Object Pascal.*

Для разработки приложений в Delphi используются специальным образом оформленные классы – компоненты.

Компонент обладает набором свойств и методов. Свойства компонента изменяются либо на этапе сборки приложения (под воздействием системы), либо программно, в процессе работы приложения (под воздействием пользователя).

Особым видом свойства компонента является событие. Процедура обработки события – это реакция приложения на изменение свойства компонента под воздействием системы или пользователя (нажатие клавиши, перемещение курсора и т.п.)

В Object Pascal объекты существуют только в динамической памяти (т.е. переменная, являющаяся объектом, по сути является указателем на объект, и содержит адрес объекта).

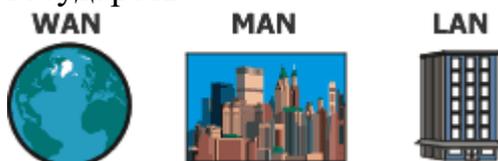
## Основные характеристики и классификация компьютерных сетей

По территориальной распространенности сети могут быть локальными, глобальными, и региональными.

Локальная сеть (LAN - Local Area Network) - сеть в пределах предприятия, учреждения, одной организации.

Региональная сеть (MAN - Metropolitan Area Network) - сеть в пределах города или области.

Глобальная сеть (WAN - Wide Area Network) – сеть на территории государства или группы государств.



По скорости передачи информации компьютерные сети делятся на низко-, средне- и высокоскоростные:

- низкоскоростные сети - до 10 Мбит/с;
- среднескоростные сети- до 100 Мбит/с;
- высокоскоростные сети - свыше 100 Мбит/с.

По типу среды передачи сети разделяются на:

- проводные (на коаксиальном кабеле, на витой паре, оптоволоконные);
- беспроводные с передачей информации по радиоканалам или в инфракрасном диапазоне.

По способу организации взаимодействия компьютеров сети делят на одноранговые и с выделенным сервером (иерархические сети).

Все компьютеры одноранговой сети равноправны. Любой пользователь сети может получить доступ к данным, хранящимся на любом компьютере.

Главное достоинство одноранговых сетей – это простота установки и эксплуатации. Главный недостаток состоит в том, что в условиях одноранговых сетей затруднено решение вопросов защиты информации. Поэтому такой способ организации сети используется для сетей с небольшим количеством компьютеров и там, где вопрос защиты данных не является принципиальным.

В иерархической сети при установке сети заранее выделяются один или несколько серверов - компьютеров, управляющих обменом данных по сети и распределением ресурсов. Любой компьютер, имеющий доступ к услугам сервера называют клиентом сети или рабочей станцией.

Сервер в иерархических сетях - это постоянное хранилище разделяемых ресурсов. Сам сервер может быть клиентом только сервера более высокого уровня иерархии. Серверы обычно представляют собой высокопроизводительные компьютеры, возможно, с несколькими параллельно работающими процессорами, винчестерами большой емкости и высокоскоростной сетевой картой.

Иерархическая модель сети является наиболее предпочтительной, так как позволяет создать наиболее устойчивую структуру сети и более

рационально распределить ресурсы. Также достоинством иерархической сети является более высокий уровень защиты данных. К недостаткам иерархической сети, по сравнению с одноранговыми сетями, относятся:

1. Необходимость дополнительной ОС для сервера.
2. Более высокая сложность установки и модернизации сети.
3. Необходимость выделения отдельного компьютера в качестве сервера

По технологии использования сервера различают сети с архитектурой файл-сервер и сети с архитектурой клиент-сервер. В первой модели используется файловый сервер, на котором хранится большинство программ и данных. По требованию пользователя ему пересылаются необходимая программа и данные. Обработка информации выполняется на рабочей станции.

В системах с архитектурой клиент-сервер обмен данными осуществляется между приложением-клиентом и приложением-сервером. Хранение данных и их обработка производится на мощном сервере, который выполняет также контроль за доступом к ресурсам и данным. Рабочая станция получает только результаты запроса.

К основным характеристикам сетей относятся:

Пропускная способность – максимальный объем данных, передаваемых сетью в единицу времени. Пропускная способность измеряется в Мбит/с.

Время реакции сети - время, затрачиваемое программным обеспечением и устройствами сети на подготовку к передаче информации по данному каналу. Время реакции сети измеряется миллисекундах.

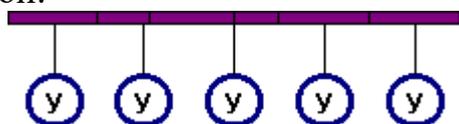
## Топология сетей

Топологией сети называется физическую или электрическую конфигурацию кабельной системы и соединений сети. В топологии сетей применяют несколько специализированных терминов:

- узел сети - компьютер, либо коммутирующее устройство сети;
- ветвь сети - путь, соединяющий два смежных узла;
- конечный узел - узел, расположенный в конце только одной ветви;
- промежуточный узел - узел, расположенный на концах более чем одной ветви;
- смежные узлы - узлы, соединенные, по крайней мере, одним путём, не содержащим никаких других узлов.

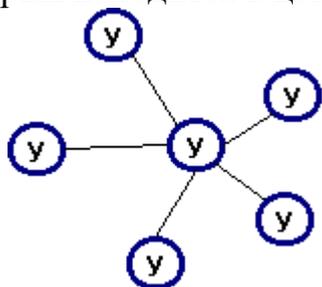
Существует всего 5 основных типов топологии сетей:

1. Топология “Общая Шина”. В этом случае подключение и обмен данными производится через общий канал связи, называемый общей шиной:



Общая шина является очень распространенной топологией для локальных сетей. Передаваемая информация может распространяться в обе стороны. Применение общей шины снижает стоимость проводки и унифицирует подключение различных модулей. Основными преимуществами такой схемы являются дешевизна и простота разводки кабеля по помещениям. Самый серьезный недостаток общей шины заключается в ее низкой надежности: любой дефект кабеля или какого-нибудь из многочисленных разъемов полностью парализует всю сеть. Другим недостатком общей шины является ее невысокая производительность, так как при таком способе подключения в каждый момент времени только один компьютер может передавать данные в сеть. Поэтому пропускная способность канала связи всегда делится здесь между всеми узлами сети.

2. Топология “Звезда”. В этом случае каждый компьютер подключается отдельным кабелем к общему устройству, называемому концентратором, который находится в центре сети:



В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. Главное преимущество этой топологии перед общей шиной - существенно большая надежность. Любые неприятности с кабелем касаются лишь того компьютера, к которому этот кабель присоединен, и только неисправность концентратора может вывести из строя всю сеть. Кроме того, концентратор может играть роль интеллектуального фильтра информации, поступающей от узлов в сеть, и при необходимости блокировать запрещенные администратором передачи.

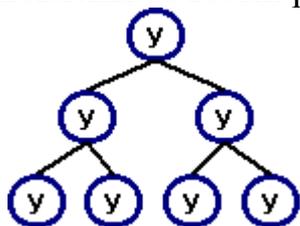
К недостаткам топологии типа звезда относится более высокая стоимость сетевого оборудования из-за необходимости приобретения концентратора. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора. В настоящее время иерархическая звезда является самым распространенным типом топологии связей как в локальных, так и глобальных сетях.

3. Топология “Кольцо”. В сетях с кольцевой топологией данные в сети передаются последовательно от одной станции к другой по кольцу, как правило, в одном направлении:



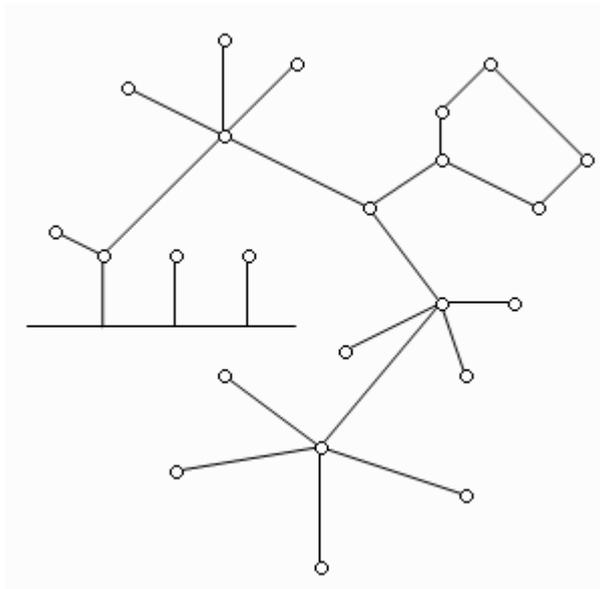
Если компьютер распознает данные как предназначенные ему, то он копирует их себе во внутренний буфер. В сети с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какой-либо станции не прервался канал связи между остальными станциями. Преимущество данной топологии - простота управления, недостаток - возможность отказа всей сети при сбое в канале между двумя узлами.

4. Ячеистая топология. Для ячеистой топологии характерна схема соединения компьютеров, при которой физические линии связи установлены со всеми рядом стоящими компьютерами:



В сети с ячеистой топологией непосредственно связываются только те компьютеры, между которыми происходит интенсивный обмен данными, а для обмена данными между компьютерами, не соединенными прямыми связями, используются транзитные передачи через промежуточные узлы. Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для глобальных сетей. Достоинства данной топологии в ее устойчивости к отказам и перегрузкам, т.к. имеется несколько способов обойти отдельные узлы.

5. Смешанная топология. В то время как небольшие сети, как правило, имеют типовую топологию - звезда, кольцо или общая шина, для крупных сетей характерно наличие произвольных связей между компьютерами. В таких сетях можно выделить отдельные произвольно подсети, имеющие типовую топологию, поэтому их называют сетями со смешанной топологией:



## Сетевое оборудование

Основными компонентами сети являются рабочие станции, серверы, передающие среды (кабели) и сетевое оборудование.

Рабочими станциями называются компьютеры сети, на которых пользователями сети реализуются прикладные задачи.

Серверы сети - это аппаратно-программные системы, выполняющие функции управления распределением сетевых ресурсов общего доступа. Сервером может быть любой подключенный к сети компьютер, на котором находятся ресурсы, используемые другими устройствами локальной сети. В качестве аппаратной части сервера используется достаточно мощные компьютеры.

Сети можно создавать с любым из типов кабеля.

1. Витая пара (TP - Twisted Pair)– это кабель, выполненный в виде скрученной пары проводов. Он может быть экранированным и неэкранированным. Экранированный кабель более устойчив к электромагнитным помехам. Витая пара наилучшим образом подходит для малых учреждений. Недостатками данного кабеля является высокий коэффициент затухания сигнала и высокая чувствительность к электромагнитным помехам, поэтому максимальное расстояние между активными устройствами в ЛВС при использовании витой пары должно быть не более 100 метров.

2. Коаксиальный кабель состоит из одного цельного или витого центрального проводника, который окружен слоем диэлектрика. Проводящий слой алюминиевой фольги, металлической оплетки или их комбинации окружает диэлектрик и служит одновременно как экран против наводок. Общий изолирующий слой образует внешнюю оболочку кабеля.

Коаксиальный кабель может использоваться в двух различных системах передачи данных: без модуляции сигнала и с модуляцией. В первом случае цифровой сигнал используется в таком виде, в каком он поступает из ПК и сразу же передается по кабелю на приемную станцию. Он имеет один канал передачи со скоростью до 10 Мбит/сек и максимальный радиус действия 4000 м. Во втором случае цифровой сигнал превращают в аналоговый и направляют его на приемную станцию, где он снова превращается в цифровой. Операция превращения сигнала выполняется модемом; каждая станция должна иметь свой модем. Этот способ передачи является многоканальным (обеспечивает передачу по десяткам каналов, используя для этого всего лишь один кабель). Таким способом можно передавать звуки, видео сигналы и другие данные. Длина кабеля может достигать до 50 км.

3. Оптоволоконный кабель является более новой технологией, используемой в сетях. Носителем информации является световой луч, который модулируется сетью и принимает форму сигнала. Такая система устойчива к внешним электрическим помехам и таким образом возможна очень быстрая, секретная и безошибочная передача данных со скоростью до 2 Гбит/с. Количество каналов в таких кабелях огромно. Передача данных выполняется только в симплексном режиме, поэтому для организации обмена данными устройства необходимо соединять двумя оптическими волокнами (на практике оптоволоконный кабель всегда имеет четное, парное кол-во волокон). К недостаткам оптоволоконного кабеля можно отнести большую стоимость, а также сложность подсоединения.

4. Радиоволны в микроволновом диапазоне используются в качестве передающей среды в беспроводных локальных сетях, либо между мостами или шлюзами для связи между локальными сетями. В первом случае максимальное расстояние между станциями составляет 200 - 300 м, во втором - это расстояние прямой видимости. Скорость передачи данных - до 2 Мбит/с.

Беспроводные локальные сети считаются перспективным направлением развития ЛС. Их преимущество - простота и мобильность. Также исчезают проблемы, связанные с прокладкой и монтажом кабельных соединений - достаточно установить интерфейсные платы на рабочие станции, и сеть готова к работе.

Выделяют следующие виды сетевого оборудования.

1. Сетевые карты – это контроллеры, подключаемые в слоты расширения материнской платы компьютера, предназначенные для передачи сигналов в сеть и приема сигналов из сети.

2. Терминаторы - это резисторы номиналом 50 Ом, которые производят затухание сигнала на концах сегмента сети.

3. Концентраторы (Hub) – это центральные устройства кабельной системы или сети физической топологии "звезда", которые при получении пакета на один из своих портов пересылает его на все остальные. В

результате получается сеть с логической структурой общей шины. Различают концентраторы активные и пассивные. Активные концентраторы усиливают полученные сигналы и передают их. Пассивные концентраторы пропускают через себя сигнал, не усиливая и не восстанавливая его.

4. Повторители (Repeater)- устройства сети, усиливает и заново формирует форму входящего аналогового сигнала сети на расстояние другого сегмента. Повторитель действует на электрическом уровне для соединения двух сегментов. Повторители ничего распознают сетевые адреса и поэтому не могут использоваться для уменьшения трафика.

5. Коммутаторы (Switch) - управляемые программным обеспечением центральные устройства кабельной системы, сокращающие сетевой трафик за счет того, что пришедший пакет анализируется для выяснения адреса его получателя и соответственно передается только ему.

Использование коммутаторов является более дорогим, но и более производительным решением. Коммутатор обычно значительно более сложное устройство и может обслуживать одновременно несколько запросов. Если по какой-то причине нужный порт в данный момент времени занят, то пакет помещается в буферную память коммутатора, где и дожидается своей очереди. Построенные с помощью коммутаторов сети могут охватывать несколько сотен машин и иметь протяженность в несколько километров.

6. Маршрутизаторы (Router)- стандартные устройства сети, работающие на сетевом уровне и позволяющее переадресовывать и маршрутизировать пакеты из одной сети в другую, а также фильтровать широковещательные сообщения.

7. Мосты (Bridge)- устройства сети, которое соединяют два отдельных сегмента, ограниченных своей физической длиной, и передают трафик между ними. Мосты также усиливают и конвертируют сигналы для кабеля другого типа. Это позволяет расширить максимальный размер сети, одновременно не нарушая ограничений на максимальную длину кабеля, количество подключенных устройств или количество повторителей на сетевой сегмент.

8. Шлюзы (Gateway) - программно-аппаратные комплексы, соединяющие разнородные сети или сетевые устройства. Шлюзы позволяет решать проблемы различия протоколов или систем адресации. Они действует на сеансовом, представительском и прикладном уровнях модели OSI.

9. Мультиплексоры – это устройства центрального офиса, которое поддерживают несколько сотен цифровых абонентских линий. Мультиплексоры посылают и получают абонентские данные по телефонным линиям, концентрируя весь трафик в одном высокоскоростном канале для передачи в Internet или в сеть компании.

10. Межсетевые экраны (firewall, брандмауэры) - это сетевые устройства, реализующие контроль за поступающей в локальную сеть и

выходящей из нее информацией и обеспечивающие защиту локальной сети посредством фильтрации информации. Большинство межсетевых экранов построено на классических моделях разграничения доступа, согласно которым субъекту (пользователю, программе, процессу или сетевому пакету) разрешается или запрещается доступ к какому-либо объекту (файлу или узлу сети) при предъявлении некоторого уникального, присущего только этому субъекту, элемента. В большинстве случаев этим элементом является пароль. В других случаях таким уникальным элементом является микропроцессорные карточки, биометрические характеристики пользователя и т. п. Для сетевого пакета таким элементом являются адреса или флаги, находящиеся в заголовке пакета, а также некоторые другие параметры.

Таким образом, межсетевой экран - это программный и/или аппаратный барьер между двумя сетями, позволяющий устанавливать только авторизованные межсетевые соединения. Обычно межсетевые экраны защищают соединяемую с Internet корпоративную сеть от проникновения извне и исключают возможность доступа к конфиденциальной информации.

### История развития Internet

В 1961 году Defence Advanced Research Agency (DARPA – оборонное агентство передовых исследовательских проектов) по заданию министерства обороны США приступило к проекту по созданию экспериментальной сети передачи пакетов. Эта сеть, названная ARPANET, предназначалась первоначально для изучения того, как поддерживать связь в случае ядерного нападения и для помощи исследователям в обмене информацией между разбросанными по всем штатам исследовательскими организациями оборонной промышленности.

В основу проекта были положены три основные идеи:

- каждый узел сети соединен с другими, так что существует несколько различных путей от узла к узлу;
- все узлы и связи рассматриваются как ненадежные;
- существуют автоматически обновляемые таблицы перенаправления пакетов - пакет, предназначенный для несоседнего узла отправляется на ближайший к нему, согласно таблице перенаправления пакетов, при недоступности этого узла - на следующий и т.д.

Созданная по таким принципам система не имела централизованного узла управления, и следовательно безболезненно могла изменять свою конфигурацию.

Эксперимент с ARPANET был настолько успешен, что многие организации захотели войти в нее с целью использования для ежедневной передачи данных. И в 1975 году ARPANET превратилась из экспериментальной сети в рабочую сеть.

В конце 80-х годов Россия подключилась к сети APRANET. В 1990 году сеть APRANET перестала существовать, и на ее месте возник Интернет. Интернет сделала возможным свободный обмен информацией, невзирая на расстояния государственные границы.

## Структура и принципы работы Интернет

Фактически, Интернет состоит из множества локальных и глобальных сетей, принадлежащих различным компаниям и предприятиям, работающих по самым разнообразным протоколам, связанных между собой различными линиями связи, физически передающих данные по телефонным проводам, оптоволокну, через спутники и радиомодемы.

За Интернет никто централизованно не платит, каждый платит за свою часть. Представители сетей собираются вместе и решают, как им соединиться друг с другом и содержать эти взаимосвязи. Пользователь платит за подключение к некоторой региональной сети, которая в свою очередь платит за свой доступ сетевому владельцу государственного масштаба. Интернет не имеет никакого собственника, здесь нет и специального органа управления, который бы контролировал всю работу сети Интернет. Локальные сети различных стран финансируются и управляются местными органами согласно политике данной страны.

Структура Интернет напоминает паутину, в узлах которой находятся компьютеры, связанные между собой линиями связи. Узлы Интернет, связанные высокоскоростными линиями связи, составляют базис Интернет. Как правило, это поставщики услуг (провайдеры). Оцифрованные данные пересылаются через маршрутизаторы, которые соединяют сети с помощью сложных алгоритмов, выбирая маршруты для информационных потоков.

Каждый компьютер в Интернет имеет свой уникальный адрес. В протоколе TCP/IP каждый компьютер адресуется четырьмя отделяемыми друг от друга точками десятичными числами, каждое из которых может иметь значение от 1 до 255. Адрес компьютера выглядит следующим образом:

19.226.192.108

Такой адрес называется IP-адресом. Этот номер может быть постоянно закреплен за компьютером или же присваиваться динамически - в тот момент, когда пользователь соединился с провайдером, но в любой момент времени в Интернет не существует двух компьютеров с одинаковыми IP-адресами.

Пользователю неудобно запоминать такие адреса, которые к тому же могут изменяться. Поэтому в Интернет существует Доменная Служба Имен (DNS - Domain Name System), которая позволяет каждый компьютер назвать по имени. В сети существуют миллионы компьютеров, и чтобы имена не повторялись, они разделены по независимым доменам.

Таким образом адрес компьютера выглядит как несколько доменов, разделенных точкой:

<сегмент n>. ... <сегмент 3>.<сегмент 2>.<сегмент 1>.

Здесь сегмент 1 – домен 1 уровня, сегмент 2 – домен 2 уровня и т.д.

Доменное имя - это уникальное имя, которое данный поставщик услуг избрал себе для идентификации, например: ic.vrn.ru или yahoo.com

Например, доменный адрес (доменное имя) www.microsoft.com обозначает компьютер с именем www в домене microsoft.com. Microsoft – это название фирмы, com - это домен коммерческих организаций. Имя компьютера www говорит о том, что на этом компьютере находится WWW-сервис. Это стандартный вид адреса серверов крупных фирм (например, www.intel.com, www.amd.com и т.д.). Имена компьютеров в разных доменах могут повторяться. Кроме того, один компьютер в сети может иметь несколько DNS-имен.

Домен 1 уровня обычно определяет страну местоположения сервера (ru – Россия; ua – Украина; uk – Великобритания; de – Германия) или вид организации (com – коммерческие организации; edu - научные и учебные организации; gov - правительственные учреждения; org – некоммерческие организации).

Когда вводится доменное имя, например, www.mrsu.ru, компьютер должен преобразовать его в адрес. Чтобы это сделать, компьютер посылает запрос серверу DNS, начиная с правой части доменного имени и двигаясь влево. Его программное обеспечение знает, как связаться с корневым сервером, на котором хранятся адреса серверов имён домена первого уровня (крайней правой части имени, например, ru). Таким образом, сервер запрашивает у корневого сервера адрес компьютера, отвечающего за домен ru. Получив информацию, он связывается с этим компьютером и запрашивает у него адрес сервера mrsu. После этого от сервера mrsu он получает адрес www компьютера, который и был целью данной прикладной программы.

Данные в Интернет пересылаются не целыми файлами, а небольшими блоками, которые называются пакетами. Каждый пакет содержит в себе адреса компьютеров отправителя и получателя, передаваемые данные и порядковый номер пакета в общем потоке данных. Благодаря тому, что каждый пакет содержит все необходимые данные, он может доставляться независимо от других, и довольно часто случается так, что пакеты добираются до места назначения разными путями. А компьютер-получатель затем выбирает из пакетов данные и собирает из них тот файл, который был заказан.

Для идентификации служб используются порты. Порт - это число, которое добавляется к адресу компьютера, которое указывает на программу, для которой данные предназначены. Каждой программе, запущенной на компьютере, соответствует определенный порт, и она реагирует только на те пакеты, которые этому порту адресованы. Существует большое количество стандартных портов, соответствующих определенным службам,

например, 21 - FTP; 23 - telnet; 25 - SMTP; 80 - HTTP; 110 - POP3; 70 - Gopher и т.д.

В Интернет используются не просто доменные имена, а универсальные указатели ресурсов URL (Universal Resource Locator).

URL включает в себя:

- метод доступа к ресурсу, т.е. протокол доступа (http, gopher, WAIS, ftp, file, telnet и др.);
- сетевой адрес ресурса (имя хост-машины и домена);
- полный путь к файлу на сервере.

В общем виде формат URL выглядит так:

method://host.domain[:port]/path/filename,

где method - одно из значений, перечисленных ниже:

file - файл на локальной системе;

http - файл на World Wide Web сервере;

gopher - файл на Gopher сервере;

wais - файл на WAIS (Wide Area Information Server) сервере;

news - группа новостей телеконференции Usenet;

telnet - выход на ресурсы сети Telnet;

ftp – файл на FTP – сервере.

host.domain – доменное имя в сети Интернет.

port - число, которое необходимо указывать, если метод требует номер порта.

Пример: <http://support.vrn.ru/archive/index.html>.

Префикс http:// указывает, что далее следует адрес Web-страницы, /archive описывает каталог с именем archiv на сервере support.vrn.ru, а index.html - имя файла.

Ниже приведены некоторые наиболее часто встречающиеся названия компьютеров сети Интернет.

Сервер в сети Интернет - это компьютер, обеспечивающий обслуживание пользователей сети: разделяемый доступ к дискам, файлам, принтеру, системе электронной почты. Обычно сервер - это совокупность аппаратного и программного обеспечения.

Сайт - обобщенное название совокупности документов в Интернет, связанных между собой ссылками.

Шлюз (gateway)- это компьютер или система компьютеров со специальным программным обеспечением, позволяющая связываться двум сетям с разными протоколами.

Домашняя страница - это персональная Web-страница конкретного пользователя или организации.

## Протоколы передачи данных

Протокол - это набор соглашений, который определяет обмен данными между различными программами. Протоколы задают способы передачи

сообщений и обработки ошибок в сети, а также позволяют разрабатывать стандарты, не привязанные к конкретной аппаратной платформе.

Сетевые протоколы предписывают правила работы компьютерам, которые подключены к сети. Они строятся по многоуровневому принципу. Протокол некоторого уровня определяет одно из технических правил связи. В настоящее время для сетевых протоколов используется модель OSI.

Модель OSI – это семиуровневая логическая модель работы сети. Модель OSI реализуется группой протоколов и правил связи, организованных в несколько уровней.

На физическом уровне определяются физические (механические, электрические, оптические) характеристики линий связи.

На канальном уровне определяются правила использования физического уровня узлами сети.

Сетевой уровень отвечает за адресацию и доставку сообщений.

Транспортный уровень контролирует очередность прохождения компонент сообщения.

Задача сеансного уровня – координация связи между двумя прикладными программами, работающими на разных рабочих станциях.

Уровень представления служит для преобразования данных из внутреннего формата компьютера в формат передачи. Прикладной уровень является пограничным между прикладной программой и другими уровнями.

Прикладной уровень обеспечивает удобный интерфейс связи сетевых программ пользователя.

Протокол TCP/IP – это два протокола нижнего уровня, являющиеся основой связи в Интернет. Протокол TCP (Transmission Control Protocol) разбивает передаваемую информацию на порции и нумерует все порции. С помощью протокола IP (Internet Protocol) все части передаются получателю. Далее с помощью протокола TCP проверяется, все ли части получены. При получении всех порций TCP располагает их в нужном порядке и собирает в единое целое.

Рассмотрим наиболее известные протоколы, используемые в сети Интернет.

HTTP (Hyper Text Transfer Protocol) – это протокол передачи гипертекста. Протокол HTTP используется при пересылке Web-страниц с одного компьютера на другой.

FTP (File Transfer Protocol) – это протокол передачи файлов со специального файлового сервера на компьютер пользователя. FTP дает возможность абоненту обмениваться двоичными и текстовыми файлами с любым компьютером сети. Установив связь с удаленным компьютером, пользователь может скопировать файл с удаленного компьютера на свой или скопировать файл со своего компьютера на удаленный.

POP (Post Office Protocol) – это стандартный протокол почтового соединения. Серверы POP обрабатывают входящую почту, а протокол POP

предназначен для обработки запросов на получение почты от клиентских почтовых программ.

Стандарт SMTP (Simple Mail Transfer Protocol) задает набор правил для передачи почты. Сервер SMTP возвращает либо подтверждение о приеме, либо сообщение об ошибке, либо запрашивает дополнительную информацию.

UUCP (Unix to Unix Copy Protocol) - это ныне устаревший, но все еще применяемый протокол передачи данных, в том числе для электронной почты. Этот протокол предполагает использование пакетного способа передачи информации, при котором сначала устанавливается соединение клиент-сервер и передается пакет данных, а затем автономно происходит его обработка, просмотр или подготовка писем.

TELNET – это протокол удаленного доступа. TELNET дает возможность абоненту работать на любой ЭВМ сети Интернет, как на своей собственной, то есть запускать программы, менять режим работы и т.д. На практике возможности лимитируются тем уровнем доступа, который задан администратором удаленной машины.

## Подключение к Интернет

Подключением к Интернет называют соединение компьютера с сервером, который постоянно подключен к Интернет.

Для подключения необходимы 5 основных составляющих:

1. Персональный компьютер.

2. Модем. Модем является устройством, обеспечивающим преобразование цифровой информации компьютера в сигналы, передаваемые по телефонной линии, и обратно. Его название произошло от основных функций - модулирование (modulation) - преобразование сигнала при передаче и демодулирование (demodulation) - обратное преобразование при приеме. Именно с помощью модема компьютер проще всего подключить к Интернет.

3. Настроенное программное обеспечение. Для подключения необходимы специальные программы, позволяющие набирать телефонный номер с помощью компьютера и осуществлять непосредственное подключение к сети Интернет (такие программы встроены в современные операционные системы).

Для установления связи компьютера с Интернет через модем требуется также пакет программ, называемый драйвером TCP/IP, для ОС Windows – это драйвер Winsock (Windows Sockets). В Windows уже имеется встроенный драйвер Winsock. Для его использования необходимо установить и настроить программу удаленного доступа к сети (Dial-Up Networking). Это может быть сделано в процессе установки Windows или позже с помощью папки “Установка и удаление программ” панели управления.

Для просмотра Web-страниц необходимо также наличие специальной программы для просмотра информации в Интернет – браузера (browser).

4. Линия связи. Обычно это аналоговая телефонная линия или линия связи ISDN (Integrated Services Digital Network), которая устанавливает цифровое соединение между персональным компьютером и провайдером. Линия ISDN может передавать информацию без предварительного преобразования в аналоговую форму с помощью модема, что увеличивает скорость обмена. Телефонная линия связи называется выделенной, если она круглосуточно соединяет провайдера и абонента для подключения к Интернет или коммутируемой, если она связывает с Интернет только во время сеанса связи.

5. Провайдер. Провайдер (ISP -Internet Service Provider)) – это поставщик услуг Интернет, организация, которая предоставляет возможность подключения к сети Интернет. Провайдер может быть местным или национальным. Он предоставляет необходимое программное обеспечение, адрес электронной почты и доступ к другим возможностям Интернет. Услуги доступа к Интернет, как правило, являются платными. Существуют различные варианты оплаты. Обычно при подключении через модем предлагаются следующие тарифы:

- почасовой, т.е. за время подключения к Интернет;
- за объем трафика (трафик- это объем информации, передаваемой по сети);
- за определенный период (сутки, месяц) без подсчета фактического времени подключения и трафика;
- за поддержание почтового ящика.

Чтобы подключить компьютер к сети Интернет в первый раз, необходимо знать:

- имя пользователя (логин);
- пароль;
- местный номер телефона провайдера;
- IP-адреса сервера DNS;
- будет ли при подключении использоваться постоянный IP- адрес или нет;
- маску IP-адреса (для локальной сети).

Настройка подключения к Интернет обеспечивается с помощью папки “Свойства обозревателя” панели управления, где следует выбрать закладку "Подключения". В закладке "Подключение" необходимо нажать кнопку "Установить...", а затем выполнить соответствующий диалог.

## WWW и HTML

WWW (World Wide Web - всемирная паутина) – это служба, предоставляющая доступ к различным ресурсам Интернет – документам, графике, аудио и видеозаписям, использующая протокол HTTP и язык HTML.

Технология WWW была разработана в 1989 г. в Женеве, в Лаборатории физики элементарных частиц Европейского центра ядерных исследований (CERN).

HTTP – это протокол передачи гипертекстовых документов. HTML (Hypertext Markup Language) – это язык разметки гипертекста. Гипертекст – это формат документа, который кроме текста, может содержать ссылки на другие гипертекстовые документы, картинки, музыку и файлы. Гиперссылки- это ссылки, позволяющие переходить от одного Web-ресурса к другому щелчком мыши. При просмотре Web-страницы в браузере ссылки выделяются визуально.

HTML – это формат гипертекстовых документов, использующихся в WWW для предоставления информации. Формат этот не описывает то, как документ должен выглядеть, а его структуру и связи. Внешний вид документа на экране пользователя определяется программой просмотра WWW - браузером. Если работать за графическим или текстовым терминалом, то в каждом случае документ будет выглядеть по-разному, но структура его останется неизменной, поскольку она задана форматом HTML. Имена файлов в формате HTML обычно имеют расширение htm, html, dhtml, shtml.

HTML – это язык тегов. Теги – это команды языка html. От остального текста они отделяются треугольными скобками. Например, <P>, </P>. Теги ставятся парами для определения начала и окончания области кода HTML, на которую они действуют. Например, <P> - открывающий тег, </P> - закрывающий тег. Теги определяют, какие параметры имеет текст в области их действия, размер, начертание шрифта, выравнивание, цвет, расположение объектов в документе и т.д..

Web-мастера - это пользователи сети, создающие web-странички и сайты. Для создания html-документов web-мастера используют визуальные редакторы (Microsoft Front Page) или простые текстовые редакторы (Блокнот Windows). Создание страниц с помощью визуальных редакторов удобнее, но редактор создает не такой оптимальный html-код, как опытный web-мастер. Кроме того, встраивать многие элементы html-кода возможно, только непосредственно редактируя код странички.

WWW работает по принципу клиент-серверы: существует множество серверов, которые по запросу клиента возвращают ему гипертекстовый документ. Чтобы использовать WWW, пользователь должен иметь специальное программное обеспечение, которое, как правило, распространяется по сети бесплатно или поставляется в комплекте с большинством других программ и услуг Интернет. Когда в браузере загружается Web-страница, то он выполняет команды, записанные на языке HTML, и выводит страничку на экран. Программные средства WWW являются универсальными для различных сервисов Интернет, а сама информационная система WWW играет интегрирующую роль.

## Браузеры

Браузер - это программа для просмотра гипертекстовых документов и перехода от одного гипертекстового документа к другому.

Основные функции браузеров следующие:

- установка связи с Web-сервером, на котором хранится документ, и загрузка всех его компонентов;
- форматирование и отображение Web-страниц в соответствии с возможностями компьютера, на котором браузер работает;
- предоставление средств для отображения мультимедийных и других объектов, входящих в состав Web-страниц, а так же механизма расширения, позволяющего настраивать программу на работу с новыми типами объектов;
- обеспечение автоматизации поиска Web-страниц и упрощение доступа к страницам, посещавшимся раньше;
- предоставление доступа к встроенным или автономным средствам для работы с другими службами Интернет, такими как электронная почта, телеконференции, FTP.

Браузеры имеют следующие кнопки управления.

Кнопка “Стоп” - предназначена для прекращения загрузки документа, если нажать её в то время, пока он ещё не загрузился. О состоянии загрузки странички можно судить по индикатору в внизу окна).

Кнопка “Назад” - предназначена для перехода на предыдущую страницу.

Кнопка “Вперёд” - предназначена для перехода обратно на страницу, на которой была нажата кнопка “Назад”.

Кнопка “Обновить” - предназначена для обновления внешнего вида страницы, т.е. её загрузки заново.

Кнопка “Печать” - предназначена для распечатки всей странички целиком, с текстом и графикой.

Кнопка “Домой” - предназначена для загрузки домашней страницы. В браузере термином “домашняя страница” (Home Page) называется первоначально загружаемая страница при запуске браузера. В качестве начальной страницы можно выбрать любую Web-страницу, находящуюся на удаленном компьютере.

Наиболее распространенными программами просмотра гипертекстовых документов на сегодняшний день являются Microsoft Интернет Explorer и Netscape Navigator.

## Поиск информации в Интернет

Поисковая система – это комплекс программ и мощных компьютеров, автоматически просматривающих ресурсы Интернет, которые они могут найти, и индексирующих их содержание. Поисковые системы могут отличаться по эффективности поиска, по языку поиска (русский,

английский и др.) и по некоторым другим возможностям. Например, одни поисковые системы находят информацию только в виде Web-страниц, другие могут просматривать и группы новостей, и файловые серверы. Результатом поиска являются гиперссылки на документы, содержащие требуемую информацию.

Наиболее известны следующие системы для поиска информации в международных информационных ресурсах:

Alta Vista (<http://www.altavista.com/>);

Google (<http://www.google.com/>);

Yahoo (<http://www.yahoo.com/>);

Infoseek (<http://www.infoseek.com/>);

Hot Bot (<http://www.hotbot.com/>).

Для поиска информации в российских информационных ресурсах:

Яндекс (<http://www.yandex.ru/>).

Рамблер (<http://www.rambler.ru/>);

Апорт (<http://www.aport.ru/>).

Для поиска информации в Интернет с использованием поисковой системы необходимо перейти на ее WEB - страницу, набрав электронный адрес или воспользоваться гипертекстовой ссылкой на эту систему.

Поисковые системы могут быть 2-х типов: универсальные и специализированные. Наиболее популярные современные поисковые системы сочетают в себе оба типа.

В универсальных системах используется обычный принцип поиска в неструктурированных документах - по ключевым словам. Ключевым словом (Keyword) документа называется отдельное слово или словосочетание, которое отражает содержание данного документа.

На начальной странице поисковой системы обычно расположено обширное меню тем и поле для ввода запроса, иногда можно задать язык для поиска. Для поиска документов по ключевому слову надо ввести это слово в поле для ввода запроса и нажать кнопку "Search" (или "Поиск" в русских системах), расположенную рядом с полем ввода. Для поиска можно использовать словосочетание; для этого надо словосочетание заключить в двойные кавычки. В некоторых системах можно осуществлять поиск по части слова, оставшаяся часть слова заменяется знаком "\*", как в шаблоне имени файла. Знаки "+" и "-" перед словом требуют обязательного присутствия или отсутствия этого слова в документе.

Существует также кнопка перехода к расширенному поиску (Advanced Search). Главное отличие расширенного поиска - использование в запросе логических операторов и круглых скобок. Для построения сложного запроса используются логические операторы AND (И), OR (ИЛИ), NOT (НЕТ) и NEAR (около; не далее чем в 10 символах). Логические операторы ставятся между словами или словосочетаниями. Здесь могут использоваться даты документов размер документов и другие критерии. Интерфейсы расширенного поиска у разных поисковых систем существенно отличаются;

для наиболее эффективного использования нужно внимательно прочитать раздел Advanced Search Help поисковой системы.

Поисковые системы обычно состоят из трех компонентов:

- поисковый робот (агент, паук или кроулер), который перемещается по сети и собирает информацию;
- база данных, которая содержит всю информацию, собираемую роботом;
- поисковый механизм, который используется как интерфейс для взаимодействия с базой данных.

Поисковые роботы - это специальные программы, которые занимаются поиском страниц в сети, извлекают гипертекстовые ссылки на этих страницах и автоматически индексируют информацию, которую они находят для построения базы данных.

При запросе к поисковой системе база данных отыскивает предмет запроса, основанный на информации, указанной в заполненной форме поиска, и выводит список ссылок на документы, соответствующих запросу. В этом списке представлены ссылки на различные Web-страницы, причем ссылки располагаются по степени убывания встреченных на данных страницах слов, совпадающих с ключевыми словами. При просмотре списка необходимо выбрать те страницы, которые нужно просмотреть. Некоторые системы составляют список ссылок по степени свежести страниц, другие же - по степени вероятности того, что данные страницы окажутся искомыми.

Специализированные справочные службы - это тематические каталоги, в которых собраны структурированные сведения об адресах серверов по той или иной тематике. Ссылки в такие каталоги заносятся не автоматически, а с помощью администраторов. Они стараются сделать свои коллекции наиболее полными, включающими все доступные ресурсы на каждую тему. В результате пользователю не нужно самому собирать все ссылки по интересующему его вопросу, а достаточно найти этот вопрос в каталоге - работа по поиску и систематизации ссылок уже сделана за него. Как правило, хорошие каталоги Интернет обеспечивают разнообразный дополнительный сервис: поиск по ключевым словам в своей базе данных, списки последних поступлений, списки наиболее интересных из них, выдачу случайной ссылки, автоматическое оповещение по электронной почте о свежих поступлениях.

Поисковые каталоги предназначены для поиска по темам. Обычно они построены по иерархическому принципу, т.е. каждый шаг поиска это выбор подраздела с более конкретной тематикой искомой информации. На нижнем уровне поиска пользователь получает относительно небольшой список ссылок на искомую информацию.

Для того чтобы обойти всю сеть, мощному роботу нужно от нескольких дней до нескольких недель. При этом составляется свежий и подробный индекс - опись доступных ресурсов. При каждом новом цикле индекс обновляется, и старые недействительные адреса удаляются. Однако

автоматизированный подход приводит к тому, что индекс оказывается засоренным большим количеством профессионально слабых, неинформативных адресов, которые пользователь нередко и получает в результате поиска.

Каталоги составляются администраторами, просматривающими каждый новый сайт прежде, чем включить его в индекс. Качество информации каталогов выше, и нетематическая информация попасть в каталог просто не может; но коллектив редакторов может не поспевать за темпами расширения Интернета. Кроме того, чем дальше, тем больше в каталоге накапливается заброшенных или устаревших адресов - его не успевают чистить. В отличие от роботов, каталоги индексируют документ не по наиболее часто встречающимся словам, а по тем ключевым словам, которые вводятся администраторами.

Если пользователя интересует хорошо разработанная и часто востребуемая тема, популярный материал, то проще воспользоваться специализированным каталогом, обычно расположенным на первой странице каждой поисковой системы. Метод работы здесь как в обычной библиотеке: двигаясь от общего к частному, достигается список нужных сайтов. Для более специального поиска, как правило, необходимо использовать ключевые слова.

Помимо услуг по нахождению сайтов, поисковые системы предоставляют широкий перечень разнообразной сопутствующей информации, например: новости, гороскопы, почтовые ящики, электронная коммерция, котировки акций, погода, спорт, географические карты, программы телевидения, лотереи и т.д.

## Электронная почта

Электронная почта (E-mail) - это служба передачи и приема текстовых сообщений или двоичных файлов в качестве вложенных с компьютера одного пользователя на компьютер другого пользователя средствами Интернет.

Электронная почта во многом удобнее обычной бумажной и имеет следующие преимущества:

- электронной почтой сообщение в большинстве случаев доставляется гораздо быстрее, чем обычной;

- электронная почта стоит дешевле, отправка сообщений электронной почты в любую страну не требует дополнительной оплаты сверх обычной оплаты за подключение к Интернет;

- для отправки письма нескольким адресатам не нужно печатать его во многих экземплярах, достаточно однажды ввести текст в компьютер;

- если нужно перечитать, исправить полученное или составленное Вами письмо, или использовать выдержки из него, это сделать легче, поскольку текст уже находится в машине;

- удобнее хранить большое количество писем на диске, чем в ящике стола; на диске их легче и искать;

- экономится бумага.

Для того, чтобы электронное письмо дошло до своего адресата, необходимо, чтобы оно было оформлено в соответствии с международными стандартами и имело стандартизованный почтовый электронный адрес. Общепринятый формат послания определяется стандартом RFC822, и имеет заголовок и непосредственно сообщение. Заголовок состоит из следующих строк (полей):

From (От): почтовый электронный адрес - от кого пришло послание;

To (Кому): почтовый электронный адрес - кому адресовано послание;

Cc (Копия): почтовые электронные адреса - кому еще направлено послание;

Subject (Тема): тема сообщения произвольной формы;

Date: дата и время отправки сообщения;

Message-Id: уникальный идентификатор послания, присвоенный ему почтовой машиной;

Reply-To: адрес абонента, куда следует отвечать на присланное письмо.

Строки заголовка From: и Date: формируются, как правило, автоматически, программными средствами. Само послание – это, как правило, текстовый файл произвольной формы.

При передаче нетекстовых данных (исполняемой программы, графической информации) применяется перекодировка сообщений, которая выполняется соответствующими программными средствами.

Кодирование информации в электронной почте определяют 2 наиболее распространенных стандарта MIME (Multipurpose Internet Mail Extensions) и Uuencode. Они позволяют вкладывать в сообщение изображения, звуковые, двоичные или сжатые файлы.

Почтовый электронный адрес имеет вид:

mailbox@site.domen,

где mailbox - идентификатор абонента. Обычно пользователь называет свой почтовый ящик тем же именем, под которым он зарегистрирован в системе. Чаще всего это имя, фамилия или фамилия с инициалами. Для того, чтобы отделить идентификатор абонента от его почтовых координат, используется значок @. Справа от знака @ располагается домен, который однозначно описывает местонахождение абонента. Составные части домена разделяются точками. Пример почтового адреса электронной почты: petrov@mrsu.ru.

В Интернет для работы с электронной почтой используется прикладные протоколы SMTP и POP. Протокол SMTP поддерживает передачу сообщений между произвольными узлами Интернет. Используя механизмы промежуточного хранения почты и повышения надежности доставки, протокол SMTP допускает использование различных транспортных служб и почтовых серверов. Почтовый протокол POP дает

конечному пользователю доступ к пришедшим к нему электронным сообщениям. POP-клиенты при запросе на получение почты требуют ввести пароль, что повышает конфиденциальность переписки.

Отправка электронных писем происходит следующим образом. SMTP-сервер определяет IP-адрес компьютера, на котором находится почтовый ящик получателя. По этому адресу SMTP-сервер отправляет письмо. Письмо получает SMTP-сервер адресата и записывает его в файл, соответствующий почтовому ящику адресата. Этот файл находится в специальной директории среди файлов владельцев других почтовых ящиков, относящихся к одному и тому же провайдеру.

Если SMTP-сервер не сумел определить, по какому IP-адресу ему отправлять письмо, или ящика, указанного в адресе не существует, или размер письма превышает ограничение, установленное провайдером на размер входящей почты, то запускается программа Mailer-Daemon, которая отправит письмо назад по адресу отправителя, приписав объяснение, почему письмо не было доставлено адресату.

При получении команды на проверку почты почтовая программа-клиент связывается с сервером входящей почты (POP-сервером) и передает ему запрос на проверку почтового ящика, содержащий имя ящика и пароль к нему. Проверив пароль, POP-сервер отправляет файлы почты почтовой программе-клиенту. Затем программа-клиент сообщает POP-серверу о том, что файл она получила, и после этого POP-сервер удаляет файл соответствующего почтового ящика.

Таким образом, для того, чтобы воспользоваться услугами электронной почты, необходимо иметь подключение к Интернет, зарегистрировать почтовый ящик на почтовом сервере и установить на компьютере программу-почтовый клиент. Наиболее популярны в настоящее время следующие почтовые клиенты: Eudora Mail, Outlook Express, Netscape Messenger, The Bat! Кроме того, многие почтовые службы позволяют получить доступ к почтовому ящику через WWW-интерфейс, т.е. в этом случае можно получить и отправить электронную почту, запустив браузер и набрав адрес web-страницы соответствующей почтовой службы.

При настройке программы-почтового клиента необходимы следующие данные:

- название ящика: название, которое будет присвоено ящику внутри почтового клиента;
- имя пользователя: имя, которое будет ставиться в поле From (От) отправляемого письма рядом с электронным адресом отправителя и которое будет видеть получатель письма;
- сервер POP3: адрес почтового сервера, с которого принимается почта (для популярной в России службы mail.ru – pop.mail.ru);
- сервер SMTP: адрес почтового сервера, на который посылается почта (для mail.ru – smtp.mail.ru);
- почтовый адрес: адрес электронной почты (например, petrov@mail.ru);

- имя пользователя для входа на POP3-сервер: уникальное на почтовом сервере имя для идентификации пользователя (в нашем случае petrov);
- пароль для получения почты.

Все эти сведения получают у провайдера Интернет или при регистрации почтового ящика.

## FTP

FTP (File Transfer Protocol - протокол передачи файлов) – это служба Интернет, реализующая протокол передачи файлов. Программы FTP-клиенты позволяют пользователю передавать файлы между двумя компьютерами, связанными между собой локальной или глобальной сетью. При этом компьютерные платформы могут быть различных типов. Протокол FTP оптимизирован именно для передачи файлов. В этом заключаются главные особенности FTP в сети.

Для получения файлов по протоколу FTP можно использовать универсальный браузер (например, Microsoft Internet Explorer или Netscape Navigator) или специальную программу (например, Cute FTP).

Практически всегда с операционной системой поставляются стандартные программы, работающие по протоколу FTP. Их исходное предназначение - передача файлов между разными компьютерами, работающими в сетях TCP/IP: на одном из компьютеров работает программа-сервер, на втором пользователь запускает программу-клиент, которая соединяется с сервером и передает или получает по протоколу FTP файлы. Пользователь зарегистрирован на обоих компьютерах и соединяется с сервером под своим именем (login) и со своим паролем (password).

Часто сервер FTP настраивают таким образом, что соединиться с ним можно будет не только под своим именем, но и под условным именем anonymous - анонимный пользователь, в качестве пароля используя адрес электронной почты. В этом случае становится доступна не вся файловая система компьютера, а некоторый набор файлов на сервере, которые составляют содержимое сервера Anonymous FTP - общедоступного файлового архива. На таких серверах сегодня доступно огромное количество различной документации и программного обеспечения.