

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**Кафедра Информатика и прикладная математика»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

**Информатика**

**Направление подготовки** 39.03.02 Социальная работа

**Профиль образовательной программы** «Социальная работа в системе социальных служб»

**Форма обучения** *очная*

## СОДЕРЖАНИЕ

<b>1. Конспект лекций .....</b>	<b>3</b>
<b>1.1 Лекция № 1 Введение в информатику .....</b>	<b>3</b>
<b>1.2 Лекция № 2 Элементы теории информации .....</b>	<b>9</b>
<b>1.3 Лекция № 3 Перевод чисел из одной системы счисления в другую систему счисления.....</b>	<b>15</b>
<b>1.4 Лекция № 4 Арифметические действия в позиционных системах счисления.....</b>	<b>18</b>
<b>1.5 Лекция № 5 Понятие и свойства алгоритма. Базовые алгоритмические структуры.....</b>	<b>19</b>
<b>1.6 Лекция № 6 Языки программирования высокого уровня.....</b>	<b>21</b>
<b>1.7 Лекция № 7 Операционные системы и среды.....</b>	<b>26</b>
<b>1.8 Лекция № 8 Операционная система Windows XP.....</b>	<b>36</b>
<b>1.9 Лекция № 9 Текстовые редакторы и процессоры.....</b>	<b>42</b>
<b>1.10 Лекция № 10 Base 2003.....</b>	<b>48</b>
<b>1.11 Лекция № 11 Классификация компьютерных сетей. Интернет как единая система ресурсов.....</b>	<b>51</b>
<b>1.12 Лекция № 12 Создание почтового ящика.....</b>	<b>57</b>
<b>2. Методические указания по выполнению лабораторных работ .....</b>	<b>63</b>
<b>3. Методические указания по проведению практических занятий .....</b>	<b>63</b>
<b>3.1 Практическое занятие № ПЗ-1 Введение в информатику .....</b>	<b>63</b>
<b>3.2 Практическое занятие № ПЗ-2 Элементы теории информации .....</b>	<b>71</b>
<b>3.3 Практическое занятие № ПЗ-3 Позиционные и непозиционные системы счисления.....</b>	<b>73</b>
<b>3.4 Практическое занятие № ПЗ-4 Перевод чисел из одной системы счисления в другую систему счисления.....</b>	<b>75</b>
<b>3.5 Практическое занятие № ПЗ-5 Понятие и свойства алгоритма. Базовые алгоритмические структуры .....</b>	<b>78</b>
<b>3.6 Практическое занятие № ПЗ-6 Языки программирования высокого уровня .....</b>	<b>82</b>
<b>3.7 Практическое занятие № ПЗ-7 Операционные системы и среды.....</b>	<b>88</b>
<b>3.8 Практическое занятие № ПЗ-8 Операционная система Windows XP.....</b>	<b>92</b>
<b>3.9 Практическое занятие № ПЗ-9 Calc2003.....</b>	<b>94</b>
<b>3.10 Практическое занятие № ПЗ- 10 Base 2003.....</b>	<b>98</b>
<b>3.11 Практическое занятие № ПЗ- 11 Классификация компьютерных сетей. Интернет как единая система ресурсов.....</b>	<b>104</b>
<b>3.12 Практическое занятие № ПЗ- 12 Создание почтового ящика.....</b>	<b>110</b>

# 1. КОНСПЕКТ ЛЕКЦИЙ

## 1.1 Лекция № 1 ( 2 часа).

**Тема:** «Введение в информатику»

### 1.1.1 Вопросы лекции:

1. Введение
2. Информационное общество
3. Понятие информатики
4. Структура информатики

**1.1.2 Краткое содержание вопросов:** *(тезисно изложить основное содержание рассматриваемых вопросов)*

#### 1. Наименование вопроса № 1

##### Введение

Хотя информатика и считается достаточно молодой наукой (по отношению ко многим другим отраслям знания), но предпосылки к ее зарождению – достаточно древние.

При рассмотрении вопроса об истории информатики будем исходить из первых признаков и событий информационного обмена, осознавая, что об информатике как о науке тогда речь не шла.

*Пример.* Первый предмет для ведения счета обнаружен в Чехии (волчья кость с зарубками) и относится к 30000 г. до н.э.

Наиболее важной и ранней предпосылкой к информационному обмену стала речь, а позже – самые первые знаковые системы (живопись, музыка, графика, танец, обряды и др.).

Затем появилась письменность: вначале она была рисуночной, иероглифической, с использованием носителей различного типа (камень, глина, дерево и т.д.).

*Пример.* В Древнем Египте около 3000 г. до н.э. появилось иероглифическое письмо на камне, а затем и иератическое (не иероглифическое) письмо на папирусе. Бронзовый век дал нам идеограммы – изображения повторяющихся систем понятий, которые в конце IV века до н.э. превратились в рисуночное, иероглифическое письмо.

Развиваются различные системы, счета и механизации (это, как известно, – предпосылка автоматизации) счета.

*Пример.* В Древнем Вавилоне около 8000 г. до н.э. использовали различные эталоны меры (каменные шары, конусы, цилиндры и т.д.). Там же около 1800 г. до н.э. начали использовать шестидесятеричную систему счисления. Древние римляне положили в основу счисления иероглифическое обозначение пальцев рук (все символы этой системы счисления можно изобразить с помощью пальцев рук). Счет на основе пальцев использовался достаточно долго и дал нам десятичную систему счисления, применяемую во всем мире.

От рисунков на камне (пиктограмм) осуществляется переход к рисункам на дощечках, глиняных пластинах (клинописи), от клинописи – к слоговому (вавилонскому) письму, от вавилонского письма – к греческому, от греческого и латинского – к основным западным письменным системам, к возникновению пунктуационного письма.

На основе латинской и греческой письменности разрабатываются терминологические системы для различных областей знания – математики, физики, медицины, химии и т.д. Развивается математический (алгебраический) язык – основа формализации различных знаний. Распространение математической символики и языка

приводит к развитию всего естествознания, так как появился адекватный и удобный аппарат для описания и исследования различных явлений.

*Пример.* Появляются символы дифференцирования, интегрирования, которые потом берутся "на вооружение" физикой, химией и другими науками.

Совершенствуются различные системы визуализации информации – карты, чертежи, пирамиды, дворцы, акведуки, механизмы и др. *Пример.* Механизмы штурма крепостей были достаточно сложны, древние водопроводные системы работают и до сих пор.

С появлением папируса повышается информационная емкость, актуализируется новое свойство информации – сжимаемость.

С появлением бумаги появляется эффективный носитель информации – книга, а изобретение печатного станка (Гуттенберга) приводит к тиражированию информации (новое свойство информационного обмена). Появляется достаточно адекватный (на тот период) инструмент массовой информационной коммуникации. Развиваются элементы виртуального мышления (например, в картинах известных художников).

Распространению информации способствует также появление и развитие библиотек, почты, университетов – центров накопления информации, знаний, культуры в обществе.

*Пример.* Появились централизованные хранилища информации, например, в столице Хеттского государства во дворце хранилось около 20 тыс. глиняных клинописных табличек.

Происходит массовое тиражирование информации, рост профессиональных знаний и развитие информационных технологий. Появляются первые признаки параллельной (по пространству и по времени) передачи и использования информации, знаний.

*Пример.* Изменение информационных свойств накладывает отпечаток и на все производство, на производственные и коммуникационные отношения, например, происходит разделение (по пространству, по времени) труда, появляется необходимость в развитии торговли, мореходства, изучении различных языков.

Дальнейший прогресс и возникновение фотографии, телеграфа, телефона, радио, кинематографа, телевидения, компьютера, компьютерной сети, сотовой связи стимулируют развитие массовых и эффективных информационных систем и технологий.

В отраслях науки формируются языковые системы: язык химических формул, язык физических законов, язык генетических связей и др..

С появлением компьютера стало возможным хранение, автоматизация и использование профессиональных знаний программ: баз данных, баз знаний, экспертных систем и т.д..

*Пример.* Персональный компьютер впервые становится средством и стимулятором автоформализации знаний и перехода от "кастового" использования ЭВМ (исключительно "кастой программистов") к общему, "пользовательскому" использованию.

Информатика от "бумажной" стадии своего развития переходит к "безбумажной", электронной стадии развития и использования.

В конце двадцатого века возник так называемый информационный кризис, "информационный взрыв", который проявился в резком росте объема научно-технических публикаций. Возникли большие сложности восприятия, переработки информации, выделения нужной информации из общего потока и др. В этих условиях появилась необходимость в едином и доступном мировом информационном пространстве, в развитии методов и технологии информатики, в развитии информатики как методологии актуализации информации, в формировании базовых технологий и систем и пересмотре роли информатики в обществе, науке, технологии.

Мир, общество начали рассматриваться с информационных позиций. Это время лавинообразного увеличения объема информации в обществе, ускорения их применения на практике, повышения требований к актуальности, достоверности, устойчивости информации. XXI век можно считать веком "информационного сообщества", единого и доступного мирового информационного пространства (поля), которое будет постоянно улучшать как производительные силы и производственные отношения, так и человеческую личность, общество.

Появление информатики как науки базируется на индустрии сбора, обработки, передачи, использования информации, на продуктах развития математики, физики, управления, техники, лингвистики, военной науки и других наук.

## 2. Наименование вопроса № 2

### Информационное общество

#### *Информационное общество*

Современное общество характеризуется резким ростом объемов информации, циркулирующей во всех сферах человеческой деятельности. Это привело к информатизации общества.

Под **информатизацией общества** понимают организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав физических и юридических лиц на основе формирования и использования **информационных ресурсов** - документов в различной форме представления.

Целью информатизации является создание **информационного общества**, когда большинство людей занято производством, хранением, переработкой и реализацией информации. Для решения этой задачи возникают новые направления в научной и практической деятельности членов общества. Так возникла информатика и информационные технологии.

#### **Характерными чертами информационного общества** являются:

1. решена проблема информационного кризиса, когда устранено противоречие между информационной лавиной и информационным голодом;
2. обеспечен приоритет информации перед другими ресурсами;
3. главная форма развития общества - информационная экономика;
4. в основу общества закладывается автоматизированная генерация, хранение, обработка и использование знаний с помощью новейшей информационной техники и технологии;
5. информационные технологии приобретают глобальный характер, охватывая все сферы социальной деятельности человека;
6. формируется информационное единство всей человеческой цивилизации;
7. с помощью средств информатики реализован свободный доступ каждого человека к информационным ресурсам всей цивилизации;
8. реализованы гуманистические принципы управления обществом и воздействия на окружающую среду.

Помимо перечисленных положительных результатов процесса информатизации общества, возможны и негативные тенденции, сопровождающие этот процесс:

1. все большее влияние приобретают средства массовой информации;
2. информационные технологии могут разрушить частную жизнь человека;
3. существенное значение приобретает проблема качественного отбора достоверной информации;

4. некоторые люди испытывают сложности адаптации к информационному обществу. Информация в информационном обществе стала стратегическим ресурсом, ибо она определяет ключевые системы общества, системы, обеспечивающие жизнедеятельность, жизнеспособность общества.

Информатизация страны состоит в информатизации в частности следующих основных систем общества (перечень неполный, хотя и охватывает все основные системы).

1. Банковских систем.

*Пример.* Виртуальные, компьютерные расчеты и платежи, прогноз банковского кредитного риска и надежности банков, разработка и использование АРМ банковского работника и др.

2. Систем рыночной экономики.

*Пример.* Прогноз и анализ спроса и предложения на рынке, моделирование поведения сегментов рынка и прибыли от продаж, разработка и использование АРМ работника рыночной экономики и др.

3. Систем социального обеспечения.

*Пример.* Прогноз и анализ инфляции в страховании, моделирование принятия решений в различных социо-экономических и социо-культурных ситуациях, в частности катастрофических; разработка и использование АРМ социального работника и др.

4. Систем налоговой службы.

*Пример.* Прогноз и анализ собираемости налогов, моделирование и прогнозирование тяжести налогового бремени, расчет оптимальных ставок налогообложения, разработка и использование АРМ работника налоговой службы и др.

3. Наименование вопроса № 3

Понятие информатики

***Что такое информатика?***

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

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

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

В 1978 году международный научный конгресс официально закрепил за понятием "**информатика**" области, связанные с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая компьютеры и их программное обеспечение, а также организационные, коммерческие, административные и социально-политические аспекты компьютеризации — массового внедрения компьютерной техники во все области жизни людей.

Таким образом, информатика базируется на компьютерной технике и немыслима без нее.

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

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

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

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

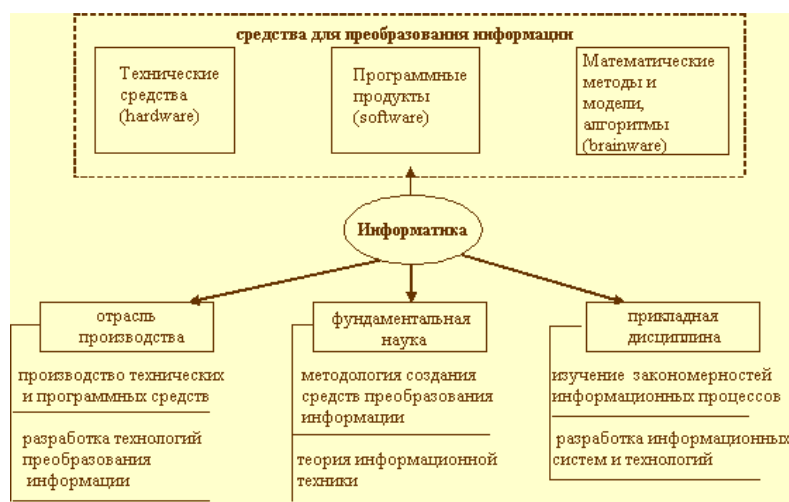
Для обозначения **программных средств**, под которыми понимается **совокупность всех программ, используемых компьютерами, и область деятельности по их созданию и применению**, используется слово **Software** (буквально — "мягкие изделия"), которое подчеркивает равнозначность самой машины и программного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

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

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

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

#### 4. Наименование вопроса № 4 Структура информатики



К **программным средствам** (продуктам) относятся операционные системы, интегрированные оболочки, системы программирования и проектирования программных продуктов, различные прикладные пакеты, такие, как текстовые и графические редакторы, бухгалтерские и издательские системы и т.д. Конкретное применение каждого программного продукта специфично и служит для решения определенного круга задач прикладного или системного характера.

**Математические методы, модели и алгоритмы** являются тем базисом, который положен в основу проектирования и изготовления любого программного или технического средства в силу их исключительной сложности и, как следствие, невозможности умозрительного подхода к созданию.

Перечисленные выше три ресурсных компонента информатики играют разную роль в процессе информатизации общества. Так, совокупность программных и технических средств, имеющихся в том или ином обществе, и позволяет сделать его информационным, когда каждый член общества имеет возможность получить практически любую (исключая, естественно, секретную) интересующую его информацию (такие потребители информации называются конечными пользователями). В то же время, сложность технических и программных систем заставляет использовать имеющиеся технические и программные продукты, а также нужные методы, модели и алгоритмы для проектирования и производства новых и совершенствования старых технических и программных систем. В этом случае можно сказать, что средства преобразования информации используются для производства себе подобных. Тогда их пользователем является специалист в области информатики, а не конечный пользователь.

Разработкой абстрактных методов, моделей и алгоритмов, а также связанных с ними математических теорий занимается **фундаментальная наука**. Ее прерогативой является исследование процессов преобразования информации и на основе этих исследований разработка соответствующих теорий, моделей, методов и алгоритмов, которые затем применяются на практике.

Практическое использование результатов исследований информатики как фундаментальной науки воплощается в информатике - **отрасли производства**. В самом деле, широко известны западные фирмы по производству программных продуктов, такие как Microsoft, Lotus, Borland, и технических средств - IBM, Apple, Intel, Hewlett Packard и другие. Помимо производства самих технических и программных средств разрабатываются также и технологии преобразования информации.

Подготовкой специалистов в области преобразования информации занимается информатика как **прикладная дисциплина**. Она изучает закономерности протекания информационных процессов в конкретных областях и методологии разработки конкретных информационных систем и технологий.



Таким образом, главная функция информатики состоит в разработке методов и средств преобразования информации с использованием компьютера, а также в применении их при организации технологического процесса преобразования информации. Это и обусловило структуру настоящего учебного пособия: информация, компьютер и информационный процесс - вот понятия, определившие структуру учебного пособия.

Выполняя свою функцию, информатика решает следующие задачи:

- исследует информационные процессы в социальных системах;
- разрабатывает информационную технику и создает новейшие технологии преобразования информации на основе результатов, полученных в ходе исследования информационных процессов;
- решает научные и инженерные проблемы создания, внедрения и обеспечения эффективного использования компьютерной техники и технологии во всех сферах человеческой деятельности.

В рамках прикладной дисциплины информатики изучаются следующие вопросы:

- понятие информации, ее свойства, измерение информации, использование в управлении;
- способы кодирования информации;
- понятие и составные части информационных процессов;
- организация технических устройств преобразования информации, в частности компьютера;
- структура и методология проектирования программного обеспечения.

## **1. 2 Лекция № 2 ( 2 часа).**

**Тема:** «Элементы теории информации»

### **1.2.1 Вопросы лекции:**

1. Понятие информации
2. Информационные процессы
3. Свойства информации
4. Вероятностный подход к измерению информации
5. Алфавитный подход к измерению информации

**1.2.2 Краткое содержание вопросов:** *(тезисно изложить основное содержание рассматриваемых вопросов)*

1. Наименование вопроса № 1  
Понятие информации.

Как видно из определения информатики, ее функций и задач, одним из ключевых понятий информатики является информация. Строгое определение информации отсутствует.

**Информация** - это сведения об окружающем мире (объекте, процессе, явлении, событии), которые являются объектом преобразования (включая хранение, передачу и т.д.) и используются для выработки поведения, для принятия решения, для управления или для обучения.

**Характерными чертами информации** являются следующие:

1. Это наиболее важный ресурс современного производства: он снижает потребность в земле, труде, капитале, уменьшает расход сырья и энергии.
2. Вызывает к жизни новые производства.

3. Является товаром, причем продавец информации ее не теряет после продажи.
4. Придает дополнительную ценность другим ресурсам, в частности, трудовым. Действительно, работник с высшим образованием ценится больше, чем со средним.
5. Информация может накапливаться.

Как следует из определения, с информацией всегда связывают три понятия (их взаимосвязь показана на рис. 2.1):

- **источник информации** - тот элемент окружающего мира, сведения о котором являются объектом преобразования;
- **потребитель информации** - тот элемент окружающего мира, который использует информацию;
- **сигнал** - материальный носитель, который фиксирует информацию для переноса ее от источника к потребителю.

Так, источником информации, которую в данный момент получает читатель настоящего учебного пособия, является информатика как сфера человеческой деятельности; потребителем - сам читатель, а сигналом - бумага с текстом (в этом случае говорят, что информация имеет бумажный носитель). Будучи прочитанной и запомненной студентом, данная информация приобретет еще один носитель - биологический, когда она "записывается" в память обучаемого. Очевидно, что источник и потребитель в этом случае не меняются.

## 2. Наименование вопроса № 2

### Информационные процессы

*Информационный процесс* - совокупность последовательных действий (операций), производимых над информацией (в виде данных, сведений, фактов, идей, гипотез, теорий и пр.), для получения какого-либо результата (достижения цели).

Информация проявляется именно в информационных процессах. Информационные процессы всегда протекают в каких-либо системах (социальных, социотехнических, биологических и пр.).

Наиболее обобщенными информационными процессами являются сбор, преобразование, использование информации.

К основным информационным процессам, изучаемым в курсе информатики, относятся: поиск, отбор, хранение, передача, кодирование, обработка, защита информации.

Информационные процессы, осуществляемые по определенным информационным технологиям, составляет основу информационной деятельности человека.

Компьютер является универсальным устройством для автоматизированного выполнения информационных процессов.

Люди имеют дело со многими видами информации. Общение людей друг с другом дома и в школе, на работе и на улице – это передача информации. Учительский рассказ или рассказ товарища, телевизионная передача, телеграмма, письмо, устное сообщение и т.д. – все это примеры передачи информации.

И мы уже говорили о том, что одну и ту же информацию можно передать и получить различными путями. Так, чтобы найти дорогу в музей в незнакомом городе, можно спросить прохожего, получить справку в справочном бюро, попытаться разобраться самому с помощью плана города или обратиться к путеводителю. Когда мы слушаем объяснение учителя, читаем книги или газеты, смотрим новости ТВ, посещаем музеи и выставки – в это время мы получаем информацию.

Человек хранит полученную информацию в голове. Мозг человека – огромное хранилище информации. Блокнот или записная книжка, ваш дневник, школьные тетрадки,

библиотека, музей, кассета с записями любимых мелодий, видеокассеты – все это примеры хранения информации.

Информацию можно обрабатывать: перевод текста с английского языка на русский и наоборот, вычисление суммы по заданным слагаемым, решение задачи, раскрашивание картинок или контурных карт – все это примеры обработки информации. Все вы любите в свое время раскрашивать книжки-раскраски. Оказывается, в это время вы занимались важным процессом – обработкой информации, черно-белый рисунок превращали в цветной.

Информацию можно даже терять. Допустим, Иванов Дима забыл дневник дома и поэтому записал домашнее задание на листочке. Но, играя на перемене, он сделал из него самолетик и запустил его. Придя домой, Дима не смог сделать домашнюю работу, он потерял информацию. Теперь ему нужно или попытаться вспомнить, что же ему задали, или позвонить однокласснику, чтобы получить нужную информацию, или идти в школу с невыполненным домашним заданием.

Вы уже заметили, что информацию можно получать, передавать, хранить, терять, распространять и преобразовывать (обрабатывать). Заметьте, что при распространении информации она не исчезает у того, кто ее передает: сообщив свое имя при знакомстве, вы наделяете своего нового товарища информацией – ваше имя вам по-прежнему хорошо известно.

Получение, хранение, передача и обработка информации – это информационные процессы. Роль информационных процессов в нашей жизни велика и с каждым годом становится все ощутимей. Поэтому человеческое общество нашего времени называют информационным обществом. Люди, живущие в информационном обществе, должны уметь пользоваться главным его инструментом, и в первую очередь универсальной информационной машиной – компьютером. Ее назвали так потому, что компьютер умеет хранить, передавать и обрабатывать информацию любого типа.

### 3 Наименование вопроса № 3 Свойства информации.

На свойства информации влияют как свойства данных, так и свойства методов её обработки.

1. **Объективность информации.** Понятие объективности информации относительно. Более объективной является та информация, в которую методы обработки вносят меньше субъективности. Например, в результате наблюдения фотоснимка природного объекта образуется более объективная информация, чем при наблюдении рисунка того же объекта. В ходе информационного процесса объективность информации всегда понижается.

2. **Полнота информации.** Полнота информации характеризует достаточность данных для принятия решения. Чем полнее данные, тем шире диапазон используемых методов их обработки и тем проще подобрать метод, вносящий минимум погрешности в информационный процесс.

3. **Адекватность информации.** Это степень её соответствия реальному состоянию дел. Неадекватная информация может образовываться при создании новой информации на основе неполных или недостоверных данных. Однако полные и достоверные данные могут приводить к созданию неадекватной информации в случае применения к ним неадекватных методов.

4. **Доступность информации.** Это мера возможности получить информацию. Отсутствие доступа к данным или отсутствие адекватных методов их обработки приводят к тому, что информация оказывается недоступной.

5. **Актуальность информации.** Это степень соответствия информации текущему моменту времени. Поскольку информационные процессы растянуты во времени, то достоверная и адекватная, но устаревшая информация может приводить к ошибочным решениям. Необходимость поиска или разработки адекватного метода обработки данных может приводить к такой задержке в получении информации, что она становится ненужной.

#### 4. Наименование вопроса № 4

##### Вероятностный подход к измерению информации

В реальной жизни существует множество ситуаций с различными вероятностями. Например, если у монеты одна сторона тяжелее другой, то при ее бросании вероятность выпадения «орла» и «решки» будет различной.

Сначала разберемся с понятием «**вероятность**». Введем следующие понятия

**испытание** - любой эксперимент;

**единичное испытание** - испытание, в котором совершается одно действие с одним предметом (например, подбрасывается монетка, или из корзины извлекается шар);

**исходы испытаний** - результаты испытания (например, при подбрасывании монеты выпал «орел», или из корзины извлекли белый шар);

**множество исходов испытания** - множество всех возможных исходов испытания;

**случайное событие** - событие, которое может произойти или не произойти (например, выигрыш билета в лотерею, извлечение карты определенной масти из колоды карт).

Для того чтобы количество информации имело положительное значение, необходимо получить сообщение о том, что произошло событие как минимум из двух равновероятных. **Такое количество информации, которое находится в сообщении о том, что произошло одно событие из двух равновероятных, принято за единицу измерения информации и равно 1 биту.**

(Огромное количество способов кодирования информации неизбежно привело пытливый ум человека к попыткам создать универсальный язык или азбуку для кодирования. Эта проблема была достаточно успешно реализована лишь в отдельных областях техники, науки и культуры. Своя система кодирования информации существует и в вычислительной технике. Она называется двоичным кодированием. Вся информацию, с которой работает вычислительная техника, можно представить в виде последовательности всего двух знаков – 1 и 0. Эти два символа называются двоичными цифрами, по-английски – binary digit или сокращенно bit – бит.

1 бит кодирует 2 понятия или сообщения (0 или 1)

2 бита – 4 разных сообщения (00 или 01 или 10 или 11)

3 бита – 8 разных сообщений.

4 бита – 16 разных сообщений и т.д.

Общая формула  $N = 2^i$ , где  $N$  – количество значений информации,  $i$  – количество бит.

Существует формула, которая связывает между собой количество возможных событий и количество информации.

$N = 2^i$ ; где  $N$  — количество возможных вариантов,  $i$  - количество информации.

Пояснение: формулы одинаковые, только применяются с разных точек зрения - кодирования и вероятности.

Если из этой формулы выразить количество информации, то получится

$$i = \log_2 N.$$

Как пользоваться этими формулами для вычислений:

- если количество возможных вариантов  $N$  является целой степенью числа 2, то производить вычисления по формуле  $N = 2^i$  достаточно легко. Вернемся к примеру:  $N = 32$ ;  $\rightarrow i = 5$ , т.к.  $32 = 2^5$ ;

- если же количество возможных вариантов информации не является целой степенью числа 2, т.е. если количество информации число вещественное, то необходимо воспользоваться калькулятором или следующей таблицей.

### Неравновероятные события

На самом деле рассмотренная нами формула является частным случаем, так как применяется только к равновероятным событиям. В жизни же мы сталкиваемся не только с равновероятными событиями, но и событиями, которые имеют разную вероятность реализации.

Например:

1. Когда сообщают прогноз погоды, то сведения о том, что будет дождь, более вероятно летом, а сообщение о снеге - зимой.

2. Если вы - лучший ученик в классе, то вероятность сообщения о том, что за контрольную работу вы получили 5, больше, чем вероятность получения двойки.

3. Если на озере живет 500 уток и 100 гусей, то вероятность подстрелить на охоте утку больше, чем вероятность подстрелить гуся.

4. Если в мешке лежат 10 белых шаров и 3 черных, то вероятность достать черный шар меньше, чем вероятность вытаскивания белого.

5. Если одна из сторон кубика будет более тяжелой, то вероятность выпадения этой стороны будет меньше, чем других сторон.

### Как вычислить количество информации в сообщении о таком событии?

Для этого необходимо использовать следующую формулу.

$i = \log_2 (1/p)$ , где  $i$  - это количество информации,  $p$  - вероятность события.

Вероятность события выражается в долях единицы и вычисляется по формуле:

$p = K / N$ , где  $K$  — величина, показывающая, сколько раз произошло интересующее нас событие,  $N$  — общее число возможных исходов какого-то процесса.

### Задача №1

В мешке находятся 20 шаров. Из них 15 белых и 5 красных. Какое количество информации несет сообщение о том, что достали: а) белый шар; б) красный шар. Сравните ответы.

Решение:

1. Найдем вероятность того, что достали белый шар:  $p_б = 15 / 20 = 0,75$ ;

2. Найдем вероятность того, что достали красный шар:  $p_к = 5 / 20 = 0,25$ .

3. Найдем количество информации в сообщении о вытаскивании белого шара:

$$i_б = \log_2 (1/p_б) = \log_2 (1/0,75) = \log_2 1,3 = 1,15470 \text{ бит.}$$

4. Найдем количество информации в сообщении о вытаскивании красного шара:  $i_к = \log_2 (1/p_к) = \log_2 (1/0,25) = \log_2 4 = 2 \text{ бит.}$

Ответ: количество информации в сообщении о том, что достали белый шар, равно 1,1547 бит. Количество информации в сообщении о том, что достали красный шар, равно 2 бит.

При сравнении ответов получается следующая ситуация: вероятность вытаскивания белого шара была больше, чем вероятность вытаскивания красного шара, а информации при этом получилось меньше. Это не случайность, а закономерная, качественная связь между вероятностью события и количеством информации в сообщении об этом событии.

### 5. Наименование вопроса № 5

Алфавитный подход к измерению информации

Познакомимся с другим способом измерения информации. Этот способ не связывает количество информации с содержанием сообщения, и называется он алфавитным подходом.

При алфавитном подходе к определению количества информации отвлекаются от содержания информации и рассматривают информационное сообщение как последовательность знаков определенной знаковой системы.

Проще всего разобраться в этом на примере текста, написанного на каком-нибудь языке. Для нас удобнее, чтобы это был русский язык.

Все множество используемых в языке символов будем традиционно называть алфавитом. Обычно под алфавитом понимают только буквы, но поскольку в тексте могут встречаться знаки препинания, цифры, скобки, то мы их тоже включим в алфавит. В алфавит также следует включить и пробел, т.е. пропуск между словами.

Полное количество символов алфавита принято называть мощностью алфавита. Будем обозначать эту величину буквой  $N$ . Например, мощность алфавита из русских букв и отмеченных дополнительных символов равна 54.

Представьте себе, что текст к вам поступает последовательно, по одному знаку, словно бумажная ленточка, выползающая из телеграфного аппарата. Предположим, что каждый появляющийся на ленте символ с одинаковой вероятностью может быть любым символом алфавита. В действительности это не совсем так, но для упрощения примем такое предположение.

В каждой очередной позиции текста может появиться любой из  $N$  символов. Тогда, согласно известной нам формуле, каждый такой символ несет  $I$  бит информации, которое можно определить из решения уравнения:  $2^I = 54$ . Получаем:  $I = 5.755$  бит.

Вот сколько информации несет один символ в русском тексте! А теперь для того, чтобы найти количество информации во всем тексте, нужно посчитать число символов в нем и умножить на  $I$ .

Посчитаем количество информации на одной странице книги. Пусть страница содержит 50 строк. В каждой строке — 60 символов. Значит, на странице умещается  $50 \times 60 = 3000$  знаков. Тогда объем информации будет равен:  $5.755 \times 3000 = 17265$  бит.

*При алфавитном подходе к измерению информации количество информации зависит не от содержания, а от размера текста и мощности алфавита.*

При использовании двоичной системы (алфавит состоит из двух знаков: 0 и 1) каждый двоичный знак несет 1 бит информации. Интересно, что сама единица измерения информации «бит» получила свое название от английского сочетания «**binary digit**» - «двоичная цифра».

Применение алфавитного подхода удобно прежде всего при использовании технических средств работы с информацией. В этом случае теряют смысл понятия «новые — старые», «понятные — непонятные» сведения. Алфавитный подход является объективным способом измерения информации в отличие от субъективного содержательного подхода.

Удобнее всего измерять информацию, когда размер алфавита  $N$  равен целой степени двойки. Например, если  $N=16$ , то каждый символ несет 4 бита информации потому, что  $2^4 = 16$ . А если  $N=32$ , то один символ «весит» 5 бит.

Ограничения на максимальный размер алфавита теоретически не существует. Однако есть алфавит, который можно назвать достаточным. С ним мы скоро встретимся при работе с компьютером. Это алфавит мощностью 256 символов. В алфавит такого размера можно поместить все практически необходимые символы: латинские и русские буквы, цифры, знаки арифметических операций, всевозможные скобки, знаки препинания....

Поскольку  $256 = 2^8$ , то один символ этого алфавита «весит» 8 бит. Причем 8 бит информации — это настолько характерная величина, что ей даже присвоили свое название — *байт*.

**1 байт = 8 бит.**

Сегодня очень многие люди для подготовки писем, документов, статей, книг и пр. используют компьютерные текстовые редакторы. Компьютерные редакторы, в основном, работают с алфавитом размером 256 символов.

В этом случае легко подсчитать объем информации в тексте. Если 1 символ алфавита несет 1 байт информации, то надо просто сосчитать количество символов; полученное число даст информационный объем текста в байтах.

Пусть небольшая книжка, сделанная с помощью компьютера, содержит 150 страниц; на каждой странице — 40 строк, в каждой строке — 60 символов. Значит страница содержит  $40 \times 60 = 2400$  байт информации. Объем всей информации в книге:  $2400 \times 150 = 360\,000$  байт.

В любой системе единиц измерения существуют основные единицы и производные от них.

Для измерения больших объемов информации используются следующие производные от байта единицы:

1 килобайт = 1Кб =  $2^{10}$  байт = 1024 байта.

1 мегабайт = 1Мб =  $2^{10}$  Кб = 1024 Кб.

1 гигабайт = 1Гб =  $2^{10}$  Мб = 1024 Мб.

### 1.3 Лекция № 3 ( 2 часа).

**Тема:** «Перевод чисел из одной системы счисления в другую систему счисления»

#### 1.3.1 Вопросы лекции:

1. Перевод чисел из любой системы счисления в десятичную систему счисления
2. Перевод чисел из десятичной системы счисления в любую систему счисления
3. Перевод чисел в системах счисления с кратными основаниями

**1.3.2 Краткое содержание вопросов:** (*тезисно изложить основное содержание рассматриваемых вопросов*)

#### 1. Наименование вопроса № 1

Перевод чисел из любой системы счисления в десятичную систему счисления

При переводе числа из двоичной (восьмеричной, шестнадцатеричной) системы в десятичную надо это число представить в виде суммы степеней основания его системы счисления.

Примеры:

Разряды	3	2	1	0	-1
Число	1	0	1	1	$1_2 = 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 11,5_{10}$
Разряды	2	1	0	-1	
Число	2	7	6	$5_8 = 2 \cdot 8^2 + 7 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} = 190,625_{10}$	
Разряды	2	1	0		
Число	1	F	$3_{16} = 1 \cdot 16^2 + 15 \cdot 16^1 + 3 \cdot 16^0 = 499_{10}$		

## 2. Наименование вопроса № 2

Перевод чисел из десятичной системы счисления в любую систему счисления.

Основание системы счисления, в которой записано число, обычно обозначается нижним индексом. Например,  $555_8$ -число записано в восьмеричной системе счисления. Если число записано в десятичной системе, то основание, как правило, не указывается. Основание системы - это тоже число, и его мы будем указывать в обычной десятичной системе.

Вообще, число  $x$  может быть представлено в системе с основанием  $p$ ,

как  $x = a_n * p^n + a_{n-1} * p^{n-1} + a_1 * p^1 + a_0 * p^0$ , где  $a_n \dots a_0$  - цифры в представлении данного числа.

Так, например,

$$1035_{10} = 1 * 10^3 + 0 * 10^2 + 3 * 10^1 + 5 * 10^0$$

$$1010_2 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 10$$

$$5617_8 = 5 * 8^3 + 6 * 8^2 + 1 * 8^1 + 7 * 8^0$$

$$A9DF_{16} = A * 16^3 + 9 * 16^2 + D * 16^1 + F * 16^0$$

Таблица соответствия чисел, записанных в различных системах счисления

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	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

1. При переводе целого десятичного числа в систему с основанием  $q$  его необходимо последовательно делить на  $q$  до тех пор, пока не останется остаток, меньший или равный  $q-1$ . Число в системе с основанием  $q$  записывается как последовательность остатков от деления, записанных в обратном порядке, начиная с последнего.

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





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

### 3 Наименование вопроса № 3

Перевод чисел в системах счисления с кратными основаниями.

При работе с компьютерами широко применяют двоичную систему счисления (поскольку на ней основано представление информации в компьютере), а также восьмеричную и шестнадцатеричную, запись в которых более компактна и удобна для человека. С другой стороны, благодаря тому что 8 и 16 — степени 2, переход между записью в двоичной и одной из этих систем осуществляется без вычислений.

Достаточно заменить каждый разряд шестнадцатеричной записи четырьмя ( $16=2^4$ ) разрядами двоичной (и наоборот) по таблице.

Аналогично происходит и перевод между двоичной и восьмеричной системой, только разряд восьмеричной соответствует трем разрядам двоичной ( $8=2^3$ ).

Следовательно, для того, чтобы перевести число из исходной системы в новую, основание которой кратно основанию исходной системы, достаточно каждую цифру переводимого числа записать при помощи **m** цифр в новой системе счисления, если основание исходной системы больше основания новой системы счисления. В противном случае каждые **m** цифр исходного числа необходимо записать при помощи одной цифры в новой системе счисления, начиная для целых чисел с младшего разряда и для правильных дробей - со старшего.

Пример.

$$[0,536]_{10} = [0,100'010'010]_2 = [0,422]_8 ; [0,1000'1001'0]_2 = [0,89]_{16}$$

$$[138]_{10} = [10'001'010]_2 = [212]_8 ; [1000'1010]_2 = [8A]_{16}$$

ДВОИЧНАЯ					ш в е с т ь н м а е д и ц а
0	0	0	0	0	
0	0	0	1	1	о
0	0	1	0	2	с
0	0	1	1	3	т
0	1	0	0	4	ь
0	1	0	1	5	н
0	1	1	0	6	м
0	1	1	1	7	а
1	0	0	0	8	е
1	0	0	1	9	д
1	0	1	0	A	и
1	0	1	1	B	ц
1	1	0	0	C	а
1	1	0	1	D	.
1	1	1	0	E	
1	1	1	1	F	

## 1.4 Лекция № 4 ( 2 часа).

**Тема:** «Арифметические действия в позиционных системах счисления»

### 1.4.1 Вопросы лекции:

1. Арифметические действия в двоичной системе счисления
2. Арифметические действия в восьмеричной и шестнадцатеричной системах счисления

**1.4.2 Краткое содержание вопросов:** (тезисно изложить основное содержание рассматриваемых вопросов)

#### 1. Наименование вопроса № 1

Арифметические действия в двоичной системе счисления.

Арифметические действия, выполняемые в двоичной системе, подчиняются тем же правилам, что и в десятичной системе. В двоичной системе счисления перенос единиц в старший разряд возникает чаще, чем в десятичной. Вот как выглядит таблица сложения в двоичной системе:

$0 + 0 = 0$	$0 + 1 = 1$
$1 + 0 = 1$	$1 + 1 = 10$ (перенос в старший разряд)

Таблица умножения для двоичных чисел еще проще:

$0 * 0 = 0$	$1 * 0 = 0$	$0 * 1 = 0$	$1 * 1 = 1$
-------------	-------------	-------------	-------------

### Пример 1

$$\begin{array}{r} + 110_2 \\ 11_2 \\ \hline 1001_2 \end{array}$$

Двоичная система счисления имеет основание 2, и для записи чисел используются всего две цифры 0 и 1 в отличие от десяти цифр десятичной системы счисления. Рассмотрим сложение одноразрядных чисел:  $0+0=0$ ,  $0+1=1$ ,  $1+0=0$ . Эти равенства справедливы как для двоичной системы, так и для десятичной системы. Чему же равно  $1+1$ ? В десятичной системе это 2. Но в двоичной системе нет цифры 2! Известно, что при десятичном сложении  $9+1$  происходит перенос 1 в старший разряд, так как старше 9 цифры нет. То есть  $9+1=10$ . В двоичной системе старшей цифрой является 1. Следовательно, в двоичной системе  $1+1=10$ , так как при сложении двух единиц происходит переполнение разряда и производится перенос в старший разряд. Переполнение разряда наступает тогда, когда значение числа в нем становится равным или большим основания. Для двоичной системы это число равно 2 ( $10_2=2_{10}$ ).

### Пример 2

$$\begin{array}{r} - 110_2 \\ 11_2 \\ \hline 11_2 \end{array}$$

Продолжая добавлять единицы, заметим:  $10_2+1=11_2$ ,  $11_2+1=100_2$  - произошла "цепная реакция", когда перенос единицы в один разряд вызывает перенос в следующий разряд. Сложение многоразрядных чисел происходит по этим же правилам с учетом возможности переносов из младших разрядов в старшие. Вычитание многоразрядных

двоичных чисел производится с учетом возможных заёмов из старших разрядов. Действия умножения и деления чисел в двоичной арифметике можно выполнять по общепринятым для позиционных систем правилам.

### Пример 3

$$\begin{array}{r} \times 110_2 \\ 11_2 \\ \hline + 110_2 \\ 110 \\ \hline 10010_2 \end{array} \quad \begin{array}{r} -110_2 \overline{)11_2} \\ 11 \ 10_2 \\ \hline 0 \end{array}$$

В основе правил арифметики любой позиционной системы лежат таблицы сложения и умножения одnorазрядных чисел

#### 2. Наименование вопроса № 2

Арифметические действия в восьмеричной и шестнадцатеричной системах счисления.

Сложить два числа:  $17_8$  и  $17_{16}$ .

Приведем число  $17_{16}$  к основанию 8 посредством двоичной системы (пробелами условно обозначено деление на тетрады и триады):  $17_{16} = 10111_2 = 10111_2 = 27_8$ .

Выполним сложение в восьмеричной системе:

$$\begin{array}{r} + 17_8 \\ 27_8 \\ \hline 46_8 \end{array}$$

Сделаем проверку, выполнив те же действия в десятичной системе:

$$17_8 = 1 \cdot 8 + 7 = 15_{10};$$

$$17_{16} = 1 \cdot 16 + 7 = 23_{10};$$

$$15 + 23 = 38_{10};$$

$$46_8 = 4 \cdot 8 + 6 = 32 + 6 = 38_{10}.$$

#### 1. 5 Лекция № 5 ( 4 часа).

**Тема:** «Понятие и свойства алгоритма. Базовые алгоритмические структуры»

##### 1.5.1 Вопросы лекции:

1. Понятие алгоритма
2. Свойства алгоритма.
3. Виды алгоритмов.
4. Способы описания алгоритмов

**1.5.2 Краткое содержание вопросов:** *(тезисно изложить основное содержание рассматриваемых вопросов)*

### 1. Наименование вопроса № 1

#### Понятие алгоритма

Алгоритмом называется точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи. Слово «алгоритм» происходит от имени математика Аль Хорезми, который сформулировал правила выполнения арифметических действий. Первоначально под алгоритмом понимали только правила выполнения четырех арифметических действий над числами. В дальнейшем это понятие стали использовать вообще для обозначения последовательности действий, приводящих к решению любой поставленной задачи. Говоря об алгоритме вычислительного процесса, необходимо понимать, что объектами, к которым применялся алгоритм, являются данные. Алгоритм решения вычислительной задачи представляет собой совокупность правил преобразования исходных данных в результатные.

### 2. Наименование вопроса № 2

#### Свойства алгоритма

Основными *свойствами* алгоритма являются:

1. детерминированность (определенность). Предполагает получение однозначного результата вычислительного процесса при заданных исходных данных. Благодаря этому свойству процесс выполнения алгоритма носит механический характер;
2. результативность. Указывает на наличие таких исходных данных, для которых реализуемый по заданному алгоритму вычислительный процесс должен через конечное число шагов остановиться и выдать искомый результат;
3. массовость. Это свойство предполагает, что алгоритм должен быть пригоден для решения всех задач данного типа;
4. дискретность. Означает расчлененность определяемого алгоритмом вычислительного процесса на отдельные этапы, возможность выполнения которых исполнителем (компьютером) не вызывает сомнений.

### 3. Наименование вопроса № 3

#### Виды алгоритмов

При всем многообразии алгоритмов решения задач в них можно выделить три основных вида вычислительных процессов:

- линейный;
- ветвящийся;
- циклический.

**Линейным** называется такой вычислительный процесс, при котором все этапы решения задачи выполняются в естественном порядке следования записи этих этапов.

**Ветвящимся** называется такой вычислительный процесс, в котором выбор направления обработки информации зависит от исходных или промежуточных данных (от результатов проверки выполнения какого-либо логического условия).

Циклом называется многократно повторяемый участок вычислений. Вычислительный процесс, содержащий один или несколько циклов, называется **циклическим**. По количеству выполнения циклы делятся на циклы с определенным (заранее заданным) числом повторений и циклы с неопределенным числом

повторений. Количество повторений последних зависит от соблюдения некоторого условия, задающего необходимость выполнения цикла. При этом условие может проверяться в начале цикла — тогда речь идет о цикле с предусловием, или в конце — тогда это цикл с постусловием.

#### 4. Наименование вопроса № 4

##### Способы описания алгоритмов

Алгоритм должен быть формализован по некоторым правилам посредством конкретных изобразительных средств. К ним относятся следующие способы записи алгоритмов: словесный, формульно-словесный, графический, язык операторных схем, алгоритмический язык.

Наибольшее распространение благодаря своей наглядности получил графический (блок-схемный) способ записи алгоритмов.

*Блок-схемой* называется графическое изображение логической структуры алгоритма, в котором каждый этап процесса обработки информации представляется в виде геометрических символов (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций. Перечень символов, их наименование, отображаемые ими функции, форма и размеры определяются ГОСТами.

#### 1. 6 Лекция № 6 ( 6 часов).

**Тема:** «Языки программирования высокого уровня»

##### 1.6.1 Вопросы лекции:

1. Языки программирования и их классификации
2. Языки низкого уровня
3. Языки программирования высокого уровня
4. Языки сверхвысокого уровня

**1.6.2 Краткое содержание вопросов:** *(тезисно изложить основное содержание рассматриваемых вопросов)*

##### 1. Наименование вопроса № 1

Языки программирования и их классификации

**Язык программирования** — формальная знаковая система, предназначенная для описания алгоритмов в форме, которая удобна для исполнителя (например, компьютера). Язык программирования определяет набор лексических, синтаксических и семантических правил, используемых при составлении компьютерной программы. Он позволяет программисту точно определить то, на какие события будет реагировать компьютер, как будут храниться и передаваться данные, а также какие именно действия следует выполнять над этими при различных обстоятельствах.

Со времени создания первых программируемых машин человечество придумало уже более двух с половиной тысяч языков программирования. Каждый год их число пополняется новыми. Некоторыми языками умеет пользоваться только небольшое число их собственных разработчиков, другие становятся известны миллионам людей. Профессиональные программисты иногда применяют в своей работе более десятка разнообразных языков программирования.

Создатели языков по-разному толкуют понятие *язык программирования*. Среди общин мест, признаваемых большинством разработчиков, находятся следующие:

- *Функция*: язык программирования предназначен для написания компьютерных программ, которые применяются для передачи компьютеру инструкций по выполнению того или иного вычислительного процесса и организации управления отдельными устройствами.
- *Задача*: язык программирования отличается от естественных языков тем, что предназначен для передачи команд и данных от человека компьютеру, в то время как естественные языки используются лишь для общения людей между собой. В принципе, можно обобщить определение "языков программирования" - это способ передачи команд, приказов, чёткого руководства к действию; тогда как человеческие языки служат также для обмена информацией.
- *Исполнение*: язык программирования может использовать специальные конструкции для определения и манипулирования структурами данных и управления процессом вычислений.

Первые языки программирования были очень примитивными и мало чем отличались от формализованных упорядоченных последовательностей единиц и нулей, понятных компьютеру. Использование таких языков было крайне неудобно с точки зрения программиста, так как он должен был знать числовые коды всех машинных команд, должен был сам распределять память под команды программы и данные.

Для того, чтобы облегчить общение человека с ЭВМ были созданы языки программирования типа **Ассемблер**. Переменные величины стали изображаться символическими именами. Числовые коды операций заменились на мнемонические обозначения, которые легче запомнить. Язык программирования приблизился к человеческому языку, и отдалился от языка машинных команд.

Языки программирования стали появляться уже с середины 50-х годов. Одним из первых языков такого типа стал язык **Фортран** (англ. FORTRAN от FORMula TRANslator – переводчик формул), разработанный в 1957 году. Фортран применяется для описания алгоритма решения научно-технических задач с помощью ЦВМ. Так же, как и первые вычислительные машины, этот язык предназначался, в основном, для проведения естественно-научных и математических расчётов. В усовершенствованном виде этот язык сохранился до нашего времени. Среди современных языков высокого уровня он является одним из наиболее используемых при проведении научных исследований. Наиболее распространены варианты Фортран-II, Фортран-IV, EASIS Fortran и их обобщения.

После Фортрана в 1958-1960 годах появился язык **Алгол** (Алгол-58, Алгол-60) (англ. ALGOL от ALGOritmic Language – алгоритмический язык). Алгол был усовершенствован в 1964-1968 годах – Алгол-68. Алгол был разработан комитетом, в который входили европейские и американские учёные. Он относится к языкам высокого уровня (high-level language) и позволяет легко переводить алгебраические формулы в программные команды. Алгол был популярен в Европе, в том числе СССР, в то время как сравнимый с ним Фортран был распространён в США и Канаде. Алгол оказал заметное влияние на все разработанные позднее языки программирования, и, в частности, на язык Pascal. Этот язык так же, как и Фортран, предназначался для решения научно-технических задач. Кроме того, этот язык применялся как средство обучения основам программирования – искусства составления программ.

Обычно под понятием Алгол подразумевается язык *Алгол-60*, в то время как *Алгол-68* рассматривается как самостоятельный язык. Даже когда язык Алгол почти перестал использоваться для программирования, он ещё оставался официальным языком для публикации алгоритмов.

В 1959 – 1960 годах был разработан язык **Кобол** (англ. COBOL от COmmon Business Oriented Language – общий язык, ориентированный на бизнес). Это язык программирования третьего поколения, предназначенный, в первую очередь, для разработки бизнес приложений. Также Кобол предназначался для решения экономических задач, обработки данных для банков, страховых компаний и других учреждений подобного рода. Разработчиком первого единого стандарта Кобола являлась Грейс Хоппер (*бабушка Кобола*).

Кобол обычно критикуется за многословность и громоздкость, поскольку одной из целей создателей языка было максимально приблизить конструкции к английскому языку. (До сих пор Кобол считается языком программирования, на котором было написано больше всего строк кода). В то же время, Кобол имел прекрасные для своего времени средства для работы со структурами данных и файлами, что обеспечило ему долгую жизнь в бизнес приложениях, по крайней мере, в США.

Почти одновременно с Коболом (1959 – 1960 гг.) в Массачусетском технологическом институте был создан язык **Лисп** (англ. LISP от LISt Processing – обработка списков). Лисп основан на представлении программы системой линейных списков символов, которые притом являются основной структурой данных языка. Лисп считается вторым после Фортрана старейшим высокоуровневым языком программирования. Этот язык широко используется для обработки символьной информации и применяется для создания программного обеспечения, имитирующего деятельность человеческого мозга.

Любая программа на Лиспе состоит из последовательности *выражений* (форм). Результат работы программы состоит в вычислении этих выражений. Все выражения записываются в виде *списков* — одной из основных структур Лиспа, поэтому они могут легко быть созданы посредством самого языка. Это позволяет создавать программы, изменяющие другие программы или макросы, позволяющие существенно расширить возможности языка.

Основной смысл Лисп-программы "жизнь" в символьном пространстве: перемещение, творчество, запоминание, создание новых миров и т.д. Лисп как метафора мозга, символ, метафора сигнала: "Как происходит биологический анализ сигналов мозгом, как внешний фактор - физическое и химическое воздействие, являющееся для организма раздражителем превращается в биологически значимый сигнал, зачастую жизненно важный, определяющий все поведение человека или животного; и как происходит разделение разных сигналов на положительные, отрицательные и безразличные, индифферентные. Сигнал это уже интегративное понятие. Он представляет собой опознавательный знак группы, комплексных раздражителей, связанных между собой общей историей и причинно следственными отношениями. В этом комплексе, системе раздражителей, сигнальный стимул сам является также составляющим элементом и при иных обстоятельствах его роль может принадлежать другому стимулу из комплекса. В сигнале концентрируется весь прошлый опыт животного или человека."

В середине 60-х годов (1963 г.) в Дартмутском колледже (США) был создан язык **Бейсик** (англ. BASIC от Beginner's Allpurpose Instruction Code – всецелевой символический код инструкций для начинающих). Со временем, когда стали появляться другие диалекты, этот «изначальный» диалект стали называть Dartmouth BASIC. Язык был основан частично на Фортран II и частично на Алгол-60, с добавлениями, делающими его удобным для работы в режиме разделения времени и, позднее, обработки текста и матричной арифметики. Первоначально Бейсик был реализован на мейнфрейме GE-265 с поддержкой множества терминалов. Вопреки распространённому убеждению, в момент своего появления это был компилируемый язык.

Бейсик был спроектирован так, чтобы студенты могли писать программы, используя терминалы с разделением времени. Он создавался как решение для проблем, связанных со сложностью более старых языков. Он предназначался для более «простых» пользователей, не столько заинтересованных в скорости программ, сколько просто в

возможности использовать компьютер для решения своих задач. В силу простоты языка Бейсик многие начинающие программисты начинают с него свой путь в программировании.

В конце 60-х – начале 70-х годов появился язык **Форт** (англ. FOURTH – четвёртый). Этот язык стал применяться в задачах управления различными системами после того, как его автор Чарльз Мур написал на нём программу, предназначенную для управления радиотелескопом Аризонской обсерватории.

Ряд свойств, а именно интерактивность, гибкость и простота разработки делают Форт весьма привлекательным и эффективным языком в прикладных исследованиях и при создании инструментальных средств. Очевидными областями применения этого языка являются встраиваемые системы управления. Также находит применение при программировании компьютеров под управлением различных операционных систем.

Появившийся в 1972 году язык **Паскаль** был назван так в честь великого французского математика XVII века, изобретателя первой в мире арифметической машины Блеза Паскаля. Этот язык был создан швейцарским учёным, специалистом в области информатики Никлаусом Виртом как язык для обучения методам программирования. Паскаль – это язык программирования общего назначения.

Особенностями языка являются строгая типизация и наличие средств структурного (процедурного) программирования. Паскаль был одним из первых таких языков. По мнению Н. Вирта, язык должен способствовать дисциплинированию программирования, поэтому, наряду со строгой типизацией, в Паскале сведены к минимуму возможные синтаксические неоднозначности, а сам синтаксис интуитивно понятен даже при первом знакомстве с языком.

Язык Паскаль учит не только тому, как правильно написать программу, но и тому, как правильно разработать метод решения задачи, подобрать способы представления и организации данных, используемых в задаче. С 1983 года язык Паскаль введён в учебные курсы информатики средних школ США.

Для обучения младших школьников Самуэлем Пайпертом был разработан язык **Лого**. Он отличается простотой и богатыми возможностями.

На основе языка Паскаль в конце 70-х годов был создан язык **Ада**, названный в честь одарённого математика Ады Лавлейс (Огасты Ады Байрон – дочери поэта Байрона). Именно она в 1843 году смогла объяснить миру возможности Аналитической машины Чарльза Бэббиджа. Язык Ада был разработан по заказу Министерства обороны США и первоначально предназначался для решения задач управления космическими полётами. Этот язык применяется в задачах управления бортовыми системами космических кораблей, системами обеспечения жизнедеятельности космонавтов в полёте, сложными техническими процессами.

Ада — это структурный, модульный, объектно-ориентированный язык программирования, содержащий высокоуровневые средства программирования параллельных процессов. Синтаксис Ады унаследован от языков типа Algol или Паскаль, но расширен, а также сделан более строгим и логичным. Ада — язык со строгой типизацией, в нём исключена работа с объектами, не имеющими типов, а автоматические преобразования типов сведены к абсолютному минимуму.

По утверждению Стефена Цейглера, разработка программного обеспечения на Аде в целом обходится на 60 % дешевле, а разработанная программа имеет в 9 раз меньше дефектов, чем при использовании языка Си.

В настоящее время популярным среди программистов является язык **Си** (C – буква английского алфавита). Язык Си берёт своё начало от двух языков - **BCPL** и **В**. В 1967 году Мартин Ричардс разработал BCPL как язык для написания системного программного обеспечения и компиляторов. В 1970 году Кен Томпсон использовал В для создания ранних версий операционной системы UNIX на компьютере DEC PDP-7. Как в BCPL, так и в В переменные не разделялись на типы - каждое значение данных занимало



одно слово в памяти и ответственность на различение, например, целых и действительных чисел целиком ложилась на плечи программиста.

Язык Си был разработан (на основе В) Деннисом Ритчи из Bell Laboratories и впервые был реализован в 1972 году на компьютере DEC PDP-11. Известность Си получил в качестве языка ОС UNIX. Сегодня практически все основные операционные системы были написаны на Си или C++. По прошествии двух десятилетий Си имеется в наличии на большинстве компьютеров. Он не зависит от аппаратной части.

В конце 70-х годов Си превратился в то, что мы называем «традиционный Си». В 1983 году Американским комитетом национальных стандартов в области компьютеров и обработки информации был учрежден единый стандарт этого языка.

Он является одним из универсальных языков программирования. В отличие от Паскаля, в нем заложены возможности непосредственного обращения к некоторым машинным командам и к определенным участкам памяти компьютера. Си широко используется как инструментальный язык для разработки операционных систем, трансляторов, баз данных и других системных и прикладных программ. Си – это язык программирования общего назначения, хорошо известный своей эффективностью, экономичностью, и переносимостью. Во многих случаях программы, написанные на Си, сравнимы по скорости с программами, написанными на языке *Ассемблера*. При этом они имеют лучшую наглядность и их более просто сопровождать. Си сочетает эффективность и мощность в относительно малом по размеру языке.

Ещё один язык, который считается языком будущего, был создан в начале 70-х годов группой специалистов Марсельского университета. Это язык **Пролог**. Своё название он получил от слов «ПРОграммирование на языке ЛОГики». В основе этого языка лежат законы математической логики. Как и язык Лисп, Пролог применяется, в основном, при проведении исследований в области программной имитации деятельности мозга человека. В отличие от описанных выше языков, этот язык не является алгоритмическим. Он относится к так называемым **дескриптивным** (от англ. descriptive – описательный) – описательным языкам. Дескриптивный язык не требует от программиста разработки всех этапов выполнения задачи. Вместо этого, в соответствии с правилами такого языка, программист должен описать базу данных, соответствующую решаемой задаче, и набор вопросов, на которые нужно получить ответы, используя данные из этой базы.

В последние десятилетия в программировании возник и получил существенное развитие **объектно-ориентированный** подход. Это метод программирования, имитирующий реальную картину мира: информация, используемая для решения задачи, представляется в виде множества взаимодействующих объектов. Каждый из объектов имеет свои свойства и способы поведения. Взаимодействие объектов осуществляется при помощи передачи сообщений: каждый объект может получать сообщения от других объектов, запоминать информацию и обрабатывать её определённым способом и, в свою очередь, посылать сообщения. Так же, как и в реальном мире, объекты хранят свои свойства и поведение вместе, наследуя часть из них от родительских объектов.

Объектно-ориентированная идеология используется во всех современных программных продуктах, включая операционные системы.

Первый объектно-ориентированный язык *Simula-67* был создан как средство моделирования работы различных приборов и механизмов. Большинство современных языков программирования – объектно-ориентированные. Среди них последние версии языка *Turbo-Pascal*, C++, *Ada* и другие.

В настоящее время широко используются системы **визуального программирования** *Visual Basic*, *Visual C++*, *Delphi* и другие. Они позволяют создавать сложные прикладные пакеты, обладающие простым и удобным пользовательским интерфейсом.

## 2. Наименование вопроса № 2

### Языки низкого уровня

В группу **языков низкого уровня** входят машинные языки и языки символического кодирования: (Автокод, Ассемблер). Операторы этого языка – это те же машинные команды, но записанные мнемоническими кодами, а в качестве операндов используются не конкретные адреса, а символические имена. Все языки низкого уровня ориентированы на определенный тип компьютера, т. е. являются машинно-зависимыми. Машинно-ориентированные языки – это языки, наборы операторов и изобразительные средства которых существенно зависят от особенностей ЭВМ (внутреннего языка, структуры памяти и т.д.).

## 3. Наименование вопроса № 3

### Языки программирования высокого уровня

Следующую, существенно более многочисленную группу составляют **языки программирования высокого уровня**. Это Фортран, Алгол, Кобол, Паскаль, Бейсик, Си, Пролог и т.д. Эти языки машинно-независимы, т.к. они ориентированы не на систему команд той или иной ЭВМ, а на систему операндов, характерных для записи определенного класса алгоритмов. Однако программы, написанные на языках высокого уровня, занимают больше памяти и медленнее выполняются, чем программы на машинных языках.

## 4. Наименование вопроса № 4

### Языки сверхвысокого уровня

К **языкам сверхвысокого уровня** можно отнести лишь Алгол-68 и APL. Повышение уровня этих языков произошло за счет введения сверхмощных операций и операторов.

Алгол-68, при разработке которого сделана попытка формализовать описание языка, приведшая к появлению абстрактной и конкретной программ. Абстрактная программа создается программистом, конкретная - выводится из первой. Предполагается, что при таком подходе принципиально невозможно породить неверную синтаксически (а в идеале и семантически) конкретную программу. Язык APL относят к языкам сверхвысокого уровня за счет введения сверхмощных операций и операторов. Запись программ на таком языке получается компактной.

*Другая классификация* делит языки на вычислительные и языки символьной обработки. К первому типу относят Фортран, Паскаль, Алгол, Бейсик, Си, ко второму типу - Лисп, Пролог, Снобол и др.

## Лекция № 7 ( 2 часа).

**Тема:** «Операционные системы и среды»

### 1.7.1 Вопросы лекции:

1. Понятие операционной системы и операционной среды
2. Операционная система как интерфейс между пользователем и компьютером
3. Операционная система как диспетчер ресурсов

4. Возможности развития операционной системы
5. Классификация операционных систем
6. Файловые системы

### **1.7.2 Краткое содержание вопросов:** *(тезисно изложить основное содержание рассматриваемых вопросов)*

#### **1. Наименование вопроса № 1**

Понятие операционной системы и операционной среды.

**Операционная система** - это набор программ, контролирующих работу прикладных программ и системных приложений и исполняющих роль интерфейса между пользователями, программистами, прикладными программами, системными приложениями и аппаратным обеспечением компьютера.

**Задача операционной системы** заключается в том, чтобы сделать использование этой вычислительной мощности доступным и по возможности удобным для пользователя. Программист может не знать детали управления конкретными ресурсами (например, диском) компьютера, а должен обращаться к операционной системе с соответствующими вызовами, чтобы получить от нее необходимые сервисы и функции. Этот набор сервисов и функций и представляет собой операционную среду, в которой выполняются прикладные программы.

Таким образом, **операционная среда** - это программная среда, образуемая операционной системой, определяющая интерфейс прикладного программирования (API) как множество системных функций и сервисов (системных вызовов), предоставляемых прикладным программам. Операционная среда может включать несколько интерфейсов прикладного программирования. Кроме основной операционной среды, называемой естественной (native), могут быть организованы путем эмуляции (моделирования) дополнительные программные среды, позволяющие выполнять приложения, которые рассчитаны на другие операционные системы и даже другие компьютеры.

Еще одно важное понятие, связанное с операционной системой, относится к реализации пользовательских интерфейсов. Как правило, любая операционная система обеспечивает удобную работу пользователя за счет средств пользовательского интерфейса. Эти средства могут быть неотъемлемой частью операционной среды, например, графический интерфейс Windows или текстовый интерфейс командной строки MS DOS, а могут быть реализованы отдельной системной программой - оболочкой операционной системы (например, Norton Commander для MS DOS). В общем случае под **оболочкой операционной системы** понимается часть операционной среды, определяющая интерфейс пользователя, его реализацию (текстовый, графический и т. п.), командные и сервисные возможности пользователя по управлению прикладными программами и компьютером.

#### **2. Наименование вопроса № 2**

Операционная система как интерфейс между пользователем и компьютером.

В области информационных технологий имеется несколько фундаментальных понятий. Одно из них – "интерфейс". Отметим, что оно может трактоваться с различных точек зрения. В предыдущей главе описано понятие "Интерфейс системных вызовов". Если искать такой термин в "Словарях" Yandex, то будет получено более десятка определений термина, большая часть которых дана в сочетаниях с другими терминами, например: "Интерфейс передачи данных", "Программный интерфейс", "Прикладной

интерфейс". В словаре "Естественные науки" на ГЛОССАРИЙ.RU дается следующее определение фундаментальному понятию.

Интерфейс в широком смысле – определенная стандартами граница между взаимодействующими независимыми объектами. Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов.

В "Издательском словаре-справочнике" есть такое определение основному термину "интерфейс". Это:

1. Система связей и взаимодействия устройств компьютера.
2. Средства взаимодействия пользователей с операционной системой компьютера, или пользовательской программой. Различают графический интерфейс пользователя (взаимодействие с компьютером организуется с помощью пиктограмм, меню, диалоговых окон и пр.) и интеллектуальный интерфейс (средства взаимодействия пользователя с компьютером на естественном языке пользователя).

Как видим, здесь этот термин имеет два значения. Но мы кратко остановимся на втором – "интерфейс пользователя". На уже упомянутом нами источнике ГЛОССАРИЙ.RU он определяется так: "Интерфейс пользователя – это элементы и компоненты программы, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением, в том числе:

- средства отображения информации, отображаемая информация, форматы и коды;
- командные режимы, язык пользователь-интерфейс;
- устройства и технологии ввода данных;
- диалоги, взаимодействие и транзакции между пользователем и компьютером;
- обратная связь с пользователем;
- поддержка принятия решений в конкретной предметной области;
- порядок использования программы и документация на нее".

По мере развития вычислительной техники методы и средства взаимодействия пользователя с операционной системой менялись. Широкое распространение цифровых вычислительных машин привело к режиму общения между человеком и ЭВМ на специальном языке. Сначала, в период пакетной обработки заданий, это реализовалось с применением специальных носителей информации (например, перфокарт, на которые наносились задания для компьютера). Но в дальнейшем, с широким распространением терминалов и клавиатуры, основным стал командный режим работы пользователя, при котором взаимодействие строилось на основе системы встроенных команд. В свободной энциклопедии "Википедия" он определен так.

Интерфейс командной строки (англ. Command line interface, CLI) – разновидность текстового интерфейса (CUI) между человеком и компьютером, в котором инструкции компьютеру даются в основном путем ввода с клавиатуры текстовых строк (команд), в UNIX-системах возможно применение мыши. Также известен под названием "консоль".

Приведем приблизительный фрагмент экрана, который появляется в режиме командной строки (рис. 1).



**Рис. 1**

Слева в строке появляется приглашение ( [asplinux@asplinuxlive ~] ), после него можно набрать команду, результаты которой выводятся далее. Приведем пример выполнения команды date в системе Linux (рис. 2).

Рис. 2

Первые операционные системы фирмы Microsoft для персональных компьютеров IBM PC (они назывались MS DOS) также поддерживали командный режим, схожий с другими системами. Строка, в которой набирались команды, была схожей с приведенными выше. Сегодня командный режим операционных систем обеспечивается эмуляторами cmd.exe (для 32-х разрядного режима) или command.com (для 16-х разрядного режима). В графическом режиме семейства UNIX/Linux командная строка эмулируется программой Терминал (xterm).

Отметим, что для компьютеров с операционной системой MS DOS удачным дополнением реализации такого интерфейса пользователя стала легендарная программа Norton Commander. Она минимизировала действия по набору текста в командной строке, позволяя оперировать, прежде всего, выбором подходящей команды из меню. В этой программе также активно используются функциональные клавиши компьютера. Википедия эту систему описывает следующим образом:

"Norton Commander (NC) – популярный файловый менеджер для DOS, первоначально разработанный американским программистом John Socha (некоторые дополнительные компоненты были полностью или частично написаны другими людьми: Linda Dudinyak – Commander Mail, вьюеры; Peter Bradeen – Commander Mail; Keith Ermel, Brian Yoder – вьюеры). Программа была выпущена компанией Peter Norton Computing (глава – Питер Нортон), которая позже была приобретена корпорацией Symantec".

Приведем пример снимка экрана этого файлового менеджера (рис. 3).



Рис. 3. Легендарный файловый менеджер Norton Commander

Популярность программы была настолько велика, что появились многочисленные клоны, которые более или менее точно копировали нортоновский интерфейс. К примеру, DOS Navigator, визуально схожий с Norton Commander-ом, предоставлял даже большие возможности. Для операционной системы Microsoft Windows появились Volkov Commander, FAR Manager, Total Commander и другие аналогичные программы. Впоследствии клоны появились и на других операционных системах: BSD, GNU/Linux – Midnight Commander, Krusader.

Norton Commander не только спровоцировал целую серию собственных клонов и реплик, но и внес в русский язык пару новых слов – "нортон" и "коммандер" стали в жаргоне пользователей ПК синонимами словосочетания "файловый менеджер".

Введенная программой парадигма работы с файлами (2 одинаковые панели, между которыми происходят операции; большинство команд выполняется по "горячим клавишам") до сих пор применяется в подавляющем большинстве файловых менеджеров.

Norton Commander также стал персонажем серий притч и анекдотов. Первая серия была написана Александром Голубевым, несколько последующих выпускались и дополнялись различными авторами, имена которых постепенно были утеряны, после чего эти рассказы перешли в состояние фольклора.

### 3 Наименование вопроса № 3

Операционная система как диспетчер ресурсов.

Компьютер представляет собой набор ресурсов, поддерживающих выполнение задач накопления, перемещения, хранения и обработки данных, а также контролирующих работу этих и других функций. Ответственность за управление этими ресурсами лежит на ОС.

ОС управляет перемещением, хранением и обработкой данных и, управляя ресурсами компьютера, контролирует его основные функции. Однако это управление имеет следующие особенности:

1. Функции операционной системы работают точно так же, как и все остальное программное обеспечение.

2. Операционная система часто передает управление другим процессам и должна ожидать, когда процессор снова позволит ей выполнять свои обязанности.

ОС — это, по сути, набор компьютерных программ. Как и любая другая программа, она отдает процессору команды. Ключевым отличием является назначение этой программы. ОС указывает процессору, как использовать другие системные ресурсы и как распределять время при исполнении других программ. Но для того, чтобы реализовать действия, предписываемые ОйСй, процессор должен приостановить работу с ней и перейти к выполнению других программ. Таким образом, ОС уступает управление процессору, чтобы он смог выполнить некоторую "полезную" работу, а затем возобновляет контроль ровно настолько, чтобы подготовить процессор к следующей части работы.

Часть операционной системы находится в основной памяти. В эту часть входит **ядро(kernel)**, содержащее основную часть наиболее часто используемых функций; там же находятся и некоторые другие компоненты операционной системы, использующиеся в данный момент времени. Остальная используемая часть основной памяти содержит другие программы и данные пользователя. Размещение этих данных в основной памяти управляется совместно операционной системой и аппаратной частью процессора, предназначенной для управления памятью. Операционная система принимает решение, когда исполняющаяся программа может использовать нужные ей устройства ввода-вывода, и управляет доступом к файлам и их использованием. Процессор также является ресурсом, поэтому операционная система должна определить, сколько времени он должен уделить исполнению той или иной пользовательской программы. В многопроцессорной системе решение должно быть принято по отношению ко всем процессорам.

### 4 Наименование вопроса № 4

Возможности развития операционной системы.

Большинство операционных систем постоянно развиваются. Происходит это в силу следующих причин.

Обновление и возникновение новых видов аппаратного обеспечения. Например, ранние версии операционных систем UNIX и OS/2 не использовали механизмы страничной организации памяти, потому что они работали на машинах, не обеспеченных соответствующими аппаратными средствами<sup>1</sup>. Более поздние версии операционных систем были доработаны таким образом, чтобы они могли использовать новые аппаратные возможности. Точно так же на устройство операционных систем повлияло использование графических терминалов и терминалов, работающих в страничном режиме, вместо алфавитно-цифровых терминалов с построчной разверткой. Такой терминал позволяет пользователю работать одновременно с несколькими приложениями в различных окнах экрана. Такая возможность требует более сложной поддержки со стороны операционной системы.

Компьютерная система

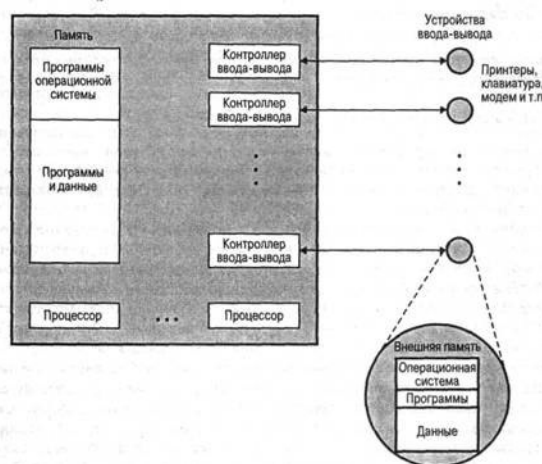


Рис. 1 Операционная система как диспетчер ресурсов

Новые сервисы. Для удовлетворения требований пользователей или нужд системных администраторов операционные системы предоставляют новые возможности. Например, если станет трудно поддерживать высокую производительность при работе с имеющимся на определенный момент инструментарием пользователя, в операционную систему могут быть добавлены новые инструменты для контроля и оценки производительности. Другой пример — поддержка новых приложений, использующих окна на экране дисплея. Эта возможность потребовала значительного обновления операционной системы.

Исправления. В каждой операционной системе есть ошибки. Время от времени они обнаруживаются и исправляются. Конечно, в исправление может вкратиться новая ошибка.

Необходимость регулярных изменений операционных систем накладывает определенные требования на их устройство. Очевидно, что эти системы должны иметь модульную конструкцию с четко определенным взаимодействием модулей; очень важную роль играет хорошая и полная документированность. Для больших программ, которыми на сегодняшний день являются типичные операционные системы, недостаточно выполнить то, что называется непосредственной модуляризацией [DENN80a] — нужно сделать нечто большее, чем простая разбивка целой программы на отдельные подпрограммы.

## 5 Наименование вопроса № 5

### Классификация операционных систем

Операционная система составляет основу программного обеспечения ПК. Операционная система представляет комплекс системных и служебных программных средств, который обеспечивает взаимодействие пользователя с компьютером и выполнение всех других программ.

С одной стороны, она опирается на базовое программное обеспечение ПК, входящее в его систему BIOS, с другой стороны, она сама является опорой для программного обеспечения более высоких уровней – прикладных и большинства служебных приложений.

Для того чтобы компьютер мог работать, на его жестком диске должна быть установлена (записана) операционная система. При включении компьютера она считывается с дисковой памяти и размещается в ОЗУ. Этот процесс называется загрузкой операционной системы.

Операционные системы различаются особенностями реализации алгоритмов управления ресурсами компьютера, областями использования.

Так, в зависимости от алгоритма управления процессором, операционные системы делятся на:

1. Однозадачные и многозадачные.
2. Однопользовательские и многопользовательские.
3. Однопроцессорные и многопроцессорные системы.
4. Локальные и сетевые.

По числу одновременно выполняемых задач операционные системы делятся на два класса:

1. Однозадачные (MS DOS).
2. Многозадачные (OS/2, Unix, Windows).

В однозадачных системах используются средства управления периферийными устройствами, средства управления файлами, средства общения с пользователями. Многозадачные ОС используют все средства, которые характерны для однозадачных, и, кроме того, управляют разделением совместно используемых ресурсов: процессор, ОЗУ, файлы и внешние устройства.

В зависимости от областей использования многозадачные ОС подразделяются на три типа:

1. Системы пакетной обработки (OS EC).
2. Системы с разделением времени (Unix, Linux, Windows).
3. Системы реального времени (RT11).

Системы пакетной обработки предназначены для решения задач, которые не требуют быстрого получения результатов. Главной целью ОС пакетной обработки является максимальная пропускная способность или решение максимального числа задач в единицу времени.

Эти системы обеспечивают высокую производительность при обработке больших объемов информации, но снижают эффективность работы пользователя в интерактивном режиме.

В системах с разделением времени для выполнения каждой задачи выделяется небольшой промежуток времени, и ни одна задача не занимает процессор надолго. Если этот промежуток времени выбран минимальным, то создается видимость одновременного выполнения нескольких задач. Эти системы обладают меньшей пропускной способностью, но обеспечивают высокую эффективность работы пользователя в интерактивном режиме.

Системы реального времени применяются для управления технологическим процессом или техническим объектом, например, летательным объектом, станком и т.д.

По числу одновременно работающих пользователей на ЭВМ ОС разделяются на однопользовательские (MS DOS) и многопользовательские (Unix, Linux, Windows 95 - XP)



В многопользовательских ОС каждый пользователь настраивает для себя интерфейс пользователя, т.е. может создать собственные наборы ярлыков, группы программ, задать индивидуальную цветовую схему, переместить в удобное место панель задач и добавить в меню Пуск новые пункты.

В многопользовательских ОС существуют средства защиты информации каждого пользователя от несанкционированного доступа других пользователей.

Многoproцессорные и однопроцессорные операционные системы. Одним из важных свойств ОС является наличие в ней средств поддержки многопроцессорной обработки данных. Такие средства существуют в OS/2, Net Ware, Widows NT. По способу организации вычислительного процесса эти ОС могут быть разделены на асимметричные и симметричные.

Одним из важнейших признаков классификации ЭВМ является разделение их на локальные и сетевые. Локальные ОС применяются на автономных ПК или ПК, которые используются в компьютерных сетях в качестве клиента.

В состав локальных ОС входит клиентская часть ПО для доступа к удаленным ресурсам и услугам. Сетевые ОС предназначены для управления ресурсами ПК включенных в сеть с целью совместного использования ресурсов. Они представляют мощные средства разграничения доступа к информации, ее целостности и другие возможности использования сетевых ресурсов.

## 6 Наименование вопроса № 6

### Файловые системы

Файловая система (англ. *file system*) — порядок, определяющий способ организации, хранения и именования данных на носителях информации ИТ-оборудования (использующего для многократной записи и хранения информации портативные флеш-карты памяти в портативных электронных устройствах: цифровых фотоаппаратах, мобильных телефонах и т. д.) и компьютерной техники. Она определяет формат содержимого и физического хранения информации, которую принято группировать в виде файлов. Конкретная файловая система определяет размер имени файла (папки), максимальный возможный размер файла и раздела, набор атрибутов файла. Некоторые файловые системы предоставляют сервисные возможности, например, разграничение доступа или шифрование файлов.

### Задачи файловой системы

Основные функции любой файловой системы нацелены на решение следующих задач:

- именование файлов;
- программный интерфейс работы с файлами для приложений;
- отображения логической модели файловой системы на физическую организацию хранилища данных;
- организация устойчивости файловой системы к сбоям питания, ошибкам аппаратных и программных средств;
- содержание параметров файла, необходимых для правильного его взаимодействия с другими объектами системы (ядро, приложения и пр.).

В многопользовательских системах появляется ещё одна задача: защита файлов одного пользователя от несанкционированного доступа другого пользователя, а также обеспечение совместной работы с файлами, к примеру, при открытии файла одним из пользователей, для других этот же файл временно будет доступен в режиме «только чтение».

## Файловые системы Windows

В Windows существуют три файловые системы **FAT**, **FAT32**, **NTFS 4.0** или **NTFS 5.0**. Какую из них выбрать? Вопрос совсем не сложный. Все зависит от Ваших потребностей или потребностей двигаться в русле со временем.

Наиболее важные параметры при выборе файловой системы:

- нужно понять для чего будет использована ФС, будь то сервер или рабочая станция

- нужно учитывать количество дисков, требования к безопасности

### FAT

Файловая система FAT может использоваться с Windows NT/2000, Windows 9x, Windows for Workgroups, MS-DOS и OS/2.

Использование файловой системы FAT является одним из лучшим выбором для томов небольшого размера, и в этом случае накладные расходы минимальны. На томах, размер которых не превышает 500 Мбайт, она работает очень хорошо. Однако на больших томах (1 Гбайт и более) FAT становится крайне неэффективной.

Для томов, размер которых находится в пределах 400—500 Мбайт, FAT является предпочтительным выбором по сравнению с NTFS, поскольку лишена накладных расходов NTFS, связанных с дисковым пространством: при форматировании тома для использования файловой системы NTFS создается целый ряд системных файлов и файл журнала транзакций, которые потребляют некоторый процент дискового пространства (и для небольших томов этот процент значителен).

### FAT32

32-разрядная файловая система FAT32 была введена с выпуском Windows 95 OSR2, и ее поддержка обеспечивается в Windows 98. Она обеспечивает оптимальный доступ к жестким дискам, повышая скорость и производительность всех операций ввода/вывода. FAT32 представляет собой усовершенствованную версию файловой системы FAT, предназначенную для использования на томах, объем которых превышает 2 Гбайт. Windows 2000 продолжает поддерживать файловую систему FAT, а также добавляет дополнительную поддержку для FAT32.

Возможности файловой системы FAT32 намного превышают возможности файловой системы FAT16. Так, эта файловая система поддерживает жесткие диски, размер которых может достигать теоретического предела 2 терабайта.

В дополнение к этому, FAT32 уменьшает размер кластера на больших дисках, снижая таким образом объем неиспользуемого пространства. Например, при использовании FAT16 на жестком диске размером 2 Гбайт, размер кластера будет составлять 32 Кбайт. Если этот же диск отформатировать с использованием FAT32, то размер кластера будет составлять только 4 Кбайт. Все утилиты Microsoft, предназначенные для работы с дисками (Format, FDISK, Defrag и ScanDisk), были переработаны для обеспечения поддержки FAT32. Кроме того, Microsoft проводит большую работу по поддержке ведущих фирм-производителей драйверов устройств и утилит для работы с диском, чтобы помочь и в обеспечении поддержки FAT32 в их продуктах. Итак, файловая система FAT32 обеспечивает следующие преимущества по сравнению с прежними реализациями файловой системы FAT:

	<i>Обеспечивается поддержка дисков размером до 2 терабайт.</i>
	Более эффективно используется дисковое пространство. За счет того, что FAT32 использует более мелкие кластеры (так, для дисков размером до 8 Гбайт используются кластеры размером по 4 Кбайт), что позволяет повысить эффективность использования дискового пространства на 10—15% по сравнению с FAT, а также снизить требования к ресурсам, необходимым для работы компьютера.
	Обеспечивается большая надежность. FAT32 обладает возможностью

	перемещения корневого каталога и использования резервной копии FAT вместо стандартной копии по умолчанию. В дополнение, загрузочная запись на дисках FAT32 была расширена, и включает в свой состав резервные копии наиболее важных структур данных. Это означает, что диски FAT32 менее чувствительны к одиночным сбоям, нежели тома FAT.
	Более быстрая загрузка программ. Благодаря тому, что FAT32 имеет меньшие размеры кластеров, приложения и необходимые для их загрузки файлы могут быть оптимальным образом размещены на диске.

## NTFS

Файловая система Windows NT (NTFS) обеспечивает такое сочетание производительности, надежности и эффективности, которое невозможно предоставить с помощью любой из реализаций FAT (как FAT16, так и FAT32). Основными целями разработки NTFS являлись обеспечение скоростного выполнения стандартных операций над файлами (включая чтение, запись, поиск) и предоставления дополнительных возможностей, включая восстановление поврежденной файловой системы на чрезвычайно больших дисках.

NTFS обладает характеристиками защищенности, поддерживая контроль доступа к данным и привилегии владельца, играющие исключительно важную роль в обеспечении целостности жизненно важных конфиденциальных данных. Папки и файлы NTFS могут иметь назначенные им права доступа вне зависимости от того, являются они общими или нет. NTFS  $\approx$  единственная файловая система в Windows NT/2000, которая позволяет назначать права доступа к отдельным файлам. Однако, если файл будет скопирован из раздела или тома NTFS в раздел или на том FAT, все права доступа и другие уникальные атрибуты, присущие NTFS, будут утрачены.

Файловая система NTFS, как и FAT, в качестве фундаментальной единицы дискового пространства использует кластеры. В NTFS размер кластера по умолчанию (когда он не задается ни командой format, ни в оснастке Управление дисками) зависит от размера тома. Если для форматирования тома NTFS используется утилита командной строки FORMAT, то нужный размер кластера можно указать в качестве параметра этой команды. Размеры кластеров по умолчанию приведены в табл. 7.3.

*Зависимость размера кластера по умолчанию от размера раздела для NTFS*

Размер раздела	Количество секторов в кластере	Размер кластера
До 512 Мбайт включительно	1	512 байт
513-1024 Мбайт (1 Гбайт)	2	1Кбайт
1025-2048 Мбайт (2 Гбайт)	4	2Кбайт
2049-4096 Мбайт (4 Гбайт)	8	4Кбайт
4097-8192 Мбайт (8 Гбайт)	16	8Кбайт
8193-16384 Мбайт (16 Гбайт)	32	16Кбайт
16385-2768 Мбайт (32 Гбайт)	64	32Кбайт
От 32 678 Мбайт	128	64Кбайт

Форматирование тома для NTFS приводит к созданию нескольких *системных файлов* и *главной таблицы файлов* (Master File Table, MFT). MFT содержит информацию обо всех файлах и папках, имеющих на томе NTFS. NTFS  $\approx$  это объектно-ориентированная файловая система, которая обрабатывает все файлы как объекты с атрибутами. Практически все объекты, существующие на томе, представляют собой файлы, а все что имеется в файле, представляет собой атрибуты  $\approx$  включая атрибуты

данных, атрибуты системы безопасности, атрибуты имени файла. Каждый занятый сектор на томе NTFS принадлежит какому-нибудь файлу. Частью файла являются даже метаданные файловой системы (информация, которая представляет собой описание самой файловой системы).

В Windows 2000 была введена новая версия NTFS  $\approx$  NTFS 5.0. Новые структуры данных, появившиеся в составе этой реализации, позволяют использовать новые возможности Windows 2000, например, квоты на использование диска для каждого пользователя, шифрование файлов, отслеживание ссылок, *точки перехода* (junction points), *встроенные наборы свойств* (native property sets). Кроме того, добавлять дополнительное дисковое пространство к томам NTFS 5.0 можно без перезагрузки. Новые возможности NTFS 5.0 приведены в табл.

*Дополнительные возможности, обеспечиваемые NTFS 4 и NTFS 5*

Функциональная возможность
Система безопасности Windows NT/2000 позволяет устанавливать различные права доступа к файлам и папкам для пользователей и групп
Ведение журнала дисковой активности позволяет быстро выполнить восстановление тома в случае сбоя подачи питания или других системных проблем
Гибкие опции форматирования позволяют более эффективно использовать дисковое пространство Windows NT/2000
Опции сжатия позволяют выполнять сжатие отдельных файлов и каталогов Тома могут расширяться и использовать дисковое пространство, не выделенное другим томам
Чередующиеся тома позволяют ускорить доступ к данным
Зеркальные тома и тома RAID-5 позволяют обеспечить отказоустойчивое хранение данных
Возможность использования файловых сервисов и сервисов печати для Macintosh (File and Print Services for Macintosh)

### Итог

NTFS обеспечивает широкий диапазон разрешений, в отличие от FAT, что дает возможность индивидуальной установки разрешений для конкретных файлов и каталогов. Это позволяет указать, какие пользователи и группы имеют доступ к файлу или папке и указать тип доступа.

Встроенные средства восстановления данных; поэтому ситуации, когда пользователь должен запускать на томе NTFS программу восстановления диска, достаточно редки. Даже в случае краха системы NTFS имеет возможность автоматически восстановить непротиворечивость файловой системы, используя журнал транзакций и информацию контрольных точек.

Реализованная в виде В-деревьев структура папок файловой системы NTFS позволяет существенно ускорить доступ к файлам в папках большого объема по сравнению со скоростью доступа к папкам такого же объема на томах FAT. NTFS позволяет осуществлять сжатие отдельных папок и файлов, можно читать сжатые файлы и писать в них без необходимости вызова программы, производящей декомпрессию.

NTFS  $\approx$  наилучший выбор для работы с томами большого объема. При этом следует учесть, что если к системе предъявляются повышенные требования (к числу которых относятся обеспечение безопасности и использование эффективного алгоритма сжатия), то часть из них можно реализовать только с помощью NTFS. Поэтому в ряде случаев нужно использовать NTFS даже на небольших томах.

## **1. 8 Лекция № 8 ( 2 часа).**

**Тема:** «Операционная система Windows XP»

### **1.8.1 Вопросы лекции:**

1. История и характеристика Windows XP
2. Варианты Windows XP
3. Графический интерфейс пользователя

### **1.8.2 Краткое содержание вопросов:** *(тезисно изложить основное содержание рассматриваемых вопросов)*

#### **1. Наименование вопроса № 1**

История и характеристика Windows XP.

На сегодняшний момент операционная система Windows фирмы Microsoft во всех проявлениях бесспорно считается самой распространенной операционной системой на ПК в мире более 150 млн. IBM PC-совместимых компьютеров, и система Windows установлена на 100 млн. из них. Очевидно что ознакомление с ПК крайне важно начинать с ознакомления с Windows, ведь без нее работа на ПК немыслима для большинства пользователей. Знание системы Windows - необходимый кирпичик в стене познания ПК.

Первые две версии Windows никуда не годились. Прорыв начался с версии Windows 3.0, она была удобна, как графическая оболочка, под которую Microsoft начали делать прикладные программы.

В это же время выпустили версию Windows NT, которая применялась в корпоративной сети, но была плоха для игр.

С выпуском версии Windows 95, компания Microsoft, стала монополистом и законодателем мод в мире программного обеспечения и персональных компьютеров.

Затем была версия Windows 98, которая была создана на базе версии Windows 95, но у продвинутых пользователей она оставила массу претензий.

Далее следовала версия Windows 2000, Windows ME, но ни одна из этих версий не пользовалась популярностью у пользователей.

И, наконец, все закончилось триумфом - Windows XP. По сути, это было продолжение версии Windows 2000, но с более интересным интерфейсом и с удобствами пользования, было одобрено и принято большинством пользователей. И по сей день, эта версия считается самой удобной в использовании.

Под именем Windows XP миру была явлена целая линейка операционных систем корпоративные ОС Windows XP Server и Windows XP Professional и домашняя Windows XP Home. А самое главное, помимо привычного 32-разрядного варианта Windows, Microsoft подготовила и 64-разрядную модификацию, предназначенную для установки на серверы, оснащенные новым 64-разрядным процессором Itanium.

Внутреннее устройство новой версии Windows, на первый взгляд, вроде бы не претерпело значительных изменений со времен Windows 2000 (если не считать традиционно улучшенной защиты системных файлов и ряда новых драйверов устройств). Одно из серьезных нововведений - встроенная система распознавания голосовых команд и голосового ввода данных (чем-то подобным, напомним, гордились еще лет шесть назад создатели OS/2).

Windows XP -- первая операционная система Microsoft с полностью настраиваемым интерфейсом! Теперь пользователи могут коренным образом изменять внешность своей ОС с помощью сменных шкур (skins), сменивших простые экранные темы времен Windows 95. Благодаря новым темам можно не только сменить рисунок на Рабочем столе, шрифт подписей к иконкам и вид курсора мыши -- в стороне не остается и обличье папок, служебных панелей и выпадающих меню. Достоинства и недостатки этой технологии

можно наблюдать уже сегодня на примере Windows Media Player, встроенного в Windows ME.

Вторым подарком Microsoft домашнему пользователю стало интеллектуальное меню Пуск, повадки которого хорошо знакомы всем пользователям Windows ME. При щелчке по этой кнопке Windows XP предлагает вам меню лишь тех программ, которыми вы часто пользуетесь, для вызова же остальных вам придется нажать на кнопку. Другие программы (More Programs).

Наконец, кардинально переработана Панель управления, -- отныне все иконки в ней свалены не абы как, а аккуратно распределены по группам.

Одним из наиболее приятных нововведений стала поддержка записи CD-R и CD-RW дисков на уровне самой ОС. Теперь вы можете работать с болванками, как с обычными дискетами, перетаскивая на чистый диск нужные файлы непосредственно в Проводнике. Конечно, это не значит, что про отдельные программы записи теперь можно забыть - копировать диски один в один, работать с режимом записи DAO и полноценной записью аудиодисков Windows по-прежнему не в состоянии.

Конечно же, теперь в составе Windows мы найдем множество новых и обновленных программ, массу мультимедийных изысков и добавлений. Нет никакого сомнения, что первую версию своей новой ОС Microsoft постарается разукрасить почище рождественской елки... И вряд ли можно их за это упрекнуть. Тем более что вместе с красивым интерфейсом пользователи наконец-то получают относительную стабильность и удобство работы. А расплачиваться за это, как водится, придется дополнительными ресурсами вашего компьютера.

Создатели Windows XP уверяют, что для установки новой ОС вам крайне важно иметь компьютер следующей конфигурации

Процессор с тактовой частотой 300 МГц или выше (рекомендуется); 233 МГц минимум.

128 мегабайт (Мб) памяти RAM или выше (рекомендуется); 64 Мб минимум, может снизить производительность и сделать недоступными некоторые функции.

1,5 гигабайта (Гб) свободного места на жестком диске.

Как обычно, эти рекомендуемые показатели можно смело умножать на два - по всем пунктам. Реально же вам крайне важно вот что

Процессор - от 800 МГц. На самом деле именно требования к процессору бывают и более либеральны -- куда важнее наличие достаточного объема оперативной памяти.

Оперативная память - от 192 Мб. Конечно, и на 128 Мб Windows XP будет работать, но без особой прыти. Идеальный же вариант - 256 Мб и выше только такой объем памяти даст возможность Windows развернуться по полной программе, не утруждая себя лихорадочными поисками свободного местечка в оперативке.

\* Свободное место на жестком диске - от 3 Гб. Сама Windows требует как минимум 2-2,5 Гб дискового пространства, а ведь нам крайне важно еще и программы устанавливать.

## 2. Наименование вопроса № 2

### Варианты Windows XP.

Windows XP выпускалась в следующих вариантах

- **Windows XP Professional Edition** была разработана для предприятий и предпринимателей и содержит такие функции, как удалённый доступ к рабочему столу

компьютера, шифрование файлов (при помощи Encrypting File System), центральное управление правами доступа и поддержка многопроцессорных систем.

- **Windows XP Home Edition** — система для домашнего применения. Выпускается как недорогая «урезанная» версия Professional Edition, но базируется на том же ядре.

- **Windows XP Tablet PC Edition** базируется на Professional Edition и содержит специальные приложения, оптимизированные для ввода данных стилусом на планшетных персональных компьютерах. Важнейшим свойством является понимание текстов, написанных от руки и адаптация графического интерфейса к поворотам дисплея. Эта версия продаётся только вместе с соответствующим компьютером.

- **Windows XP Media Center Edition** базируется на Professional Edition и содержит специальные мультимедийные приложения. Компьютер, как правило, оснащён ТВ-картой и пультом дистанционного управления (ПДУ). Важнейшим свойством является возможность подключения к телевизору и управление компьютером через ПДУ благодаря упрощённой системе управления Windows. Эта система содержит также функции для приёма УКВ-радио.

- **Windows XP Embedded** — это встраиваемая компонентная операционная система на базе Windows XP Professional Edition и предназначена для применения в различных встраиваемых системах: системах промышленной автоматизации, банкоматах, медицинских приборах, кассовых терминалах, игровых автоматах, VoIP-компонентах и т. п. Windows XP Embedded включает дополнительные функции по встраиванию, среди которых фильтр защиты от записи (EWF и FBWF), загрузка с флеш-памяти, CD-ROM, сети, использование собственной оболочки системы и т. п.

- **Windows Embedded POSReady** — специализированная операционная система на базе Windows XP Embedded, сконфигурированная для пунктов обслуживания и оптимизированная для розничной торговли и сферы услуг. На базе этой платформы можно создавать банкомат, платёжный терминал, A3C, кассовый аппарат и т. п. Дополнительно Windows Embedded for Point of Service включает технологию POS for .NET для быстрой разработки торговых приложений и поддержки торгового периферийного оборудования.

- **Windows XP Professional x64 Edition** — специальная 64-разрядная версия, разработанная для процессоров с технологией AMD64 Opteron и Athlon 64 фирмы AMD и процессоров с технологией EM64T фирмы Intel. Эта система не поддерживает процессоры других производителей, а также не работает с процессором Intel Itanium. Хотя первые 64-разрядные процессоры появились в 2003 году, Windows XP Professional x64 Edition вышла в свет только в апреле 2005 года. Основным достоинством системы является быстрая работа с большими числами (Long Integer и Double Float). Таким образом, эта система очень эффективна, например, при выполнении вычислений, использующих числа с плавающей запятой, необходимых в таких областях, как создание спецэффектов для кинофильмов и трёхмерной анимации, а также разработка технических и научных приложений. Данная система поддерживает смешанный режим, то есть одновременную работу 32- и 64-разрядных приложений, однако для этого все драйверы должны быть в 64-разрядном исполнении. Это означает, что большинство 32-разрядных приложений могут работать и в этой системе. Исключение составляют лишь те приложения, которые сильно зависят от аппаратного обеспечения компьютера, например, антивирусы и дефрагментаторы.

- **Windows XP 64-bit Edition** — это издание разрабатывалось специально для рабочих станций с архитектурой IA-64 и микропроцессорами Itanium. Это издание Windows XP более не развивается с 2005 года, после того, как HP прекратил разработку рабочих станций с микропроцессорами Itanium. Поддержка этой архитектуры осталась в серверных версиях операционной системы Windows.

- **Windows XP Edition N** — система без Windows Media Player и других мультимедиа-приложений. Эти версии созданы под давлением Европейской Антимонопольной Комиссии, которая требовала «облегчить» Windows XP. В настоящее время этот дистрибутив рассчитан на развивающиеся страны. При желании пользователь может бесплатно загрузить все недостающие приложения с веб-сайта Microsoft. Существует как в Home, так и в Professional вариантах.

- **Windows XP Starter Edition** — сильно функционально ограниченная версия для развивающихся стран и финансово слабых регионов. В этой версии возможна одновременная работа только 3 приложений, и каждое приложение может создать не более 3 окон. В системе полностью отсутствуют сетевые функции, не поддерживается высокая разрешающая способность, а также не допускается использование более 512 мегабайт оперативной памяти или жёсткого диска объёмом более 120 гигабайт. Система может работать на процессорах уровня Intel Celeron или AMD Duron.

- **Windows Fundamentals for Legacy PCs** — урезанная версия Microsoft Windows XP Embedded Service Pack 2, предназначенная для устаревших компьютеров.

### 3 Наименование вопроса № 3

#### Графический интерфейс пользователя

- Выделение в Windows Explorer осуществляется прозрачным синим прямоугольником.

- Падающая тень от ярлыков на рабочем столе.

- Боковая панель, ориентированная на выполнение задач в окне проводника («common tasks»).

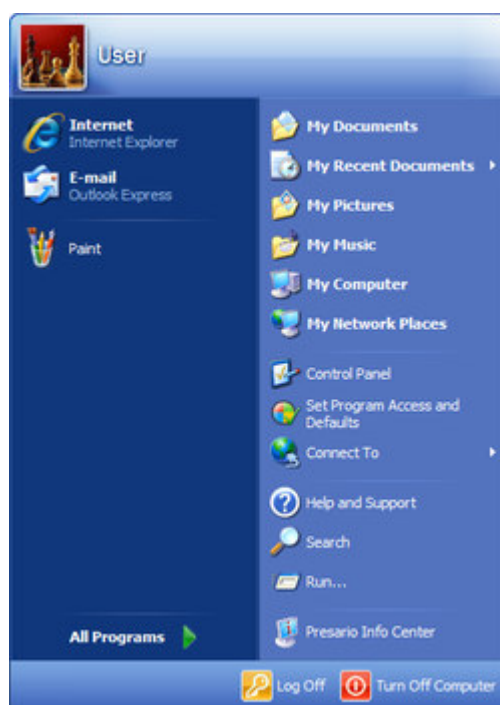
- Группирование кнопок одного приложения на панели задач в одну кнопку, при определённом количестве разных запущенных приложений, что позволяет часто избегать необходимости её «прокрутки».

- Появилась возможность заблокировать панель задач и вспомогательные панели, во избежание их случайного изменения.

- Цветовое выделение элементов в меню «Пуск», принадлежащих недавно добавленным программам.

- Меню отбрасывают тени (в Windows 2000 тень отбрасывал указатель мыши, но не элементы меню).





Windows XP анализирует производительность системы с определёнными визуальными эффектами и в зависимости от этого активирует их или нет, учитывая возможное падение или рост производительности. Пользователи также могут изменять данные параметры, используя диалоговые окна настройки, при этом можно либо гибко выбрать активность тех или иных визуальных эффектов, либо отдать это на управление системе или же выбрать максимальную производительность или лучший вид графического интерфейса.

Некоторые эффекты, такие как полупрозрачность и т.п., требуют наличия производительной графической подсистемы, на старых видеокартах производительность может сильно упасть и Microsoft рекомендует отключить эти возможности в таком случае.

В Windows XP появилась возможность использовать Visual Styles, позволяющие изменить графический интерфейс пользователя. Luna — новый стиль графического интерфейса, входящий в поставку XP и являющийся интерфейсом по умолчанию для компьютеров, имеющих более 64 мегабайт оперативной памяти. Возможно использовать и другие Visual Styles, но они должны быть подписаны цифровой подписью Microsoft (так как имеют важное значение в функционировании системы). Один из интерфейсов доступен на версии Windows XP Black Edition by Spa.

Для обхода этого ограничения некоторые пользователи используют специальное программное обеспечение, такое, как TGTSoft's StyleXP, а иногда и изменённую версию библиотеки uxtheme.dll.

Также существует стиль «классический», повторяющий стиль интерфейса Windows 2000 (который использует на 4 МБ меньше памяти, чем Luna), а также многочисленные стили, созданные сторонними разработчиками. Для версии Media Center Microsoft разработала «визуальный стиль» Royale, который включён в эту версию Windows XP и доступен для установки в других версиях XP.

Для Windows XP были созданы более 100 значков компанией The Iconfactory, известной своим набором бесплатных значков для операционной системы Mac OS X

## **1. 9 Лекция № 9 ( 2 часа).**

**Тема:** «Текстовые редакторы и процессоры»

### **1.9.1 Вопросы лекции:**

1. Офисный пакет приложений OpenOffice
2. Текстовые редакторы
3. Write – текстовый процессор
4. Параметры абзаца

**1.9.2 Краткое содержание вопросов:** *(тезисно изложить основное содержание рассматриваемых вопросов)*

#### **1. Наименование вопроса № 1**

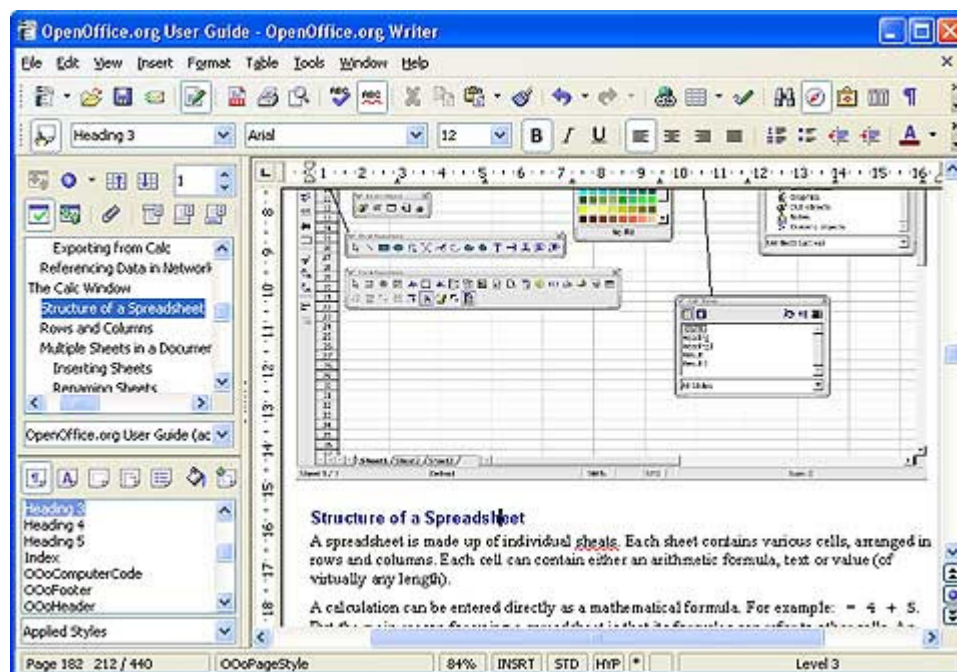
Офисный пакет приложений OpenOffice.

В последнее время о проекте OpenOffice.org заговорили практически все уважающие себя печатные и электронные издания. Мы тоже посчитали нужным внести свою лепту.

OpenOffice.org - это международный проект с открытыми исходными кодами, созданный для разработки универсального офисного пакета, работающего на разных операционных платформах, имеющего открытый API и формат файлов, основанный на XML. OpenOffice.org - это и, собственно, сам пакет офисных программ, разработанный в рамках данного проекта. В него входят: текстовый процессор, программа для работы с электронными таблицами, графический редактор, система презентаций и простенькая БД. По своим возможностям он сопоставим с аналогичными коммерческими программами и вполне может рассматриваться как альтернатива им. Ну а главным плюсом является то, что использование OpenOffice.org не требует никаких лицензионных отчислений.

Обратимся к истории, чтобы понять, откуда берет свое начало OpenOffice.org. В девяностых годах немецкая компания StarDivision выпускала офисный пакет StarOffice. Коды продукта были приобретены корпорацией Sun в 1999 году, а спустя год открыты всему миру из соображений конкурентной борьбы с Microsoft Office, таким образом предоставляя бесплатную альтернативу, к тому же основанную на открытом коде. Проект получил название OpenOffice.org. В настоящее время корпорация Sun финансирует проект OpenOffice.org и использует код OpenOffice.org как базу для своего офисного пакета StarOffice, который дополнительно включает в себя ряд частных, коммерческих технологий.

Отношение пользователей к первым версиям продукта было довольно пренебрежительное. С недостатками мирились во имя бесплатности пакета. Однако 20 октября 2005 года на сайте www.openoffice.org появился доступный для свободного скачивания релиз 2.0 этого офисного комплекта. Объем исходного кода по сравнению с первой версией увеличился в два раза! Фактически объем исправлений настолько велик, что разница между OpenOffice.org 2.0 и OpenOffice.org 1.1.0 даже больше, чем между Microsoft Office 97 и Microsoft Office 2003. Нынешний OpenOffice.org 2.0 избавился от многих недостатков, обзавелся практически всеми необходимыми компонентами и даже пытается кое в чем задавать тон. Одним словом, сегодня этот продукт интересен как никогда.



### OpenOffice.org Writer

OpenOffice.org - продукт тяжеловесный и многоплатформенный. Написан на C++ и Java, причем в новейшей серии OpenOffice.org 2.x использование Java увеличилось. Интересен тот факт, что создатели OpenOffice.org изначально взяли курс на совместимость своего продукта с форматами Microsoft Office. Более того, используя OpenOffice.org, иногда получается открыть поврежденные файлы форматов Microsoft Office, которые не открываются даже в своих "родных" программах.

OpenOffice.org в настоящее время локализован в более чем 60 странах мира, среди которых есть Россия и Украина. Сборка, предназначенная для России, имеет полностью переведенный интерфейс, словарь переносов русских слов, в скором времени ожидается перевод встроенной справки и подробного руководства пользователя. Кроме того, разработаны на основе пакета Cyrillic Tools дополнительные модули для улучшенной поддержки кириллических документов, что бывает порой необходимо, например, при работе с отчетами, сгенерированными в "1С:Бухгалтерии". Русифицированная версия пакета носит название [Ru.OpenOffice.org](http://Ru.OpenOffice.org).

Пакет включает в себя следующие программы:

- OpenOffice.org Writer - программа для работы с текстовыми документами и HTML, аналог Microsoft Word;
- OpenOffice.org Calc - программа для работы с электронными таблицами, аналог Microsoft Excel;
- OpenOffice.org Base - программа для создания баз данных;
- OpenOffice.org Draw - программа для создания и редактирования изображений;
- OpenOffice.org Impress - программа для создания презентаций, аналог Microsoft PowerPoint;
- OpenOffice.org Math - программа для работы с математическими формулами.

### 2. Наименование вопроса № 2

Текстовые редакторы.

**Writer**- это мощный, современный инструмент для создания разного рода текста.

#### Запуск редактора Writer

Щелкните мышью на кнопке Пуск и выберите в главном меню пункт Программы, а в открывшемся каскадном меню пункт **OpenOffice**.

После того как программа загрузится, на экране появится окно текстового редактора **Writer**.

По умолчанию при запуске **Writer** в его окне обычно присутствуют элементы: строка заголовка, строка меню, строка состояния, две панели инструментов, масштабная линейка, вертикальная и горизонтальная полосы прокрутки, кнопки открытия и закрытия окна.

## **ОКНО Writer**

### **Использование меню**

Строка меню, содержащая девять меню, расположена в верхней части экрана, под строкой заголовка. В названии каждого меню есть подчеркнутая буква, которую можно использовать для раскрытия соответствующего меню, нажав ее одновременно с клавишей <Alt>. Этот метод доступа к меню может пригодиться, если ваша мышь вдруг откажется работать, но удобнее всего раскрывать меню, щелкнув на его названии.

### **Панели инструментов.**

Самый быстрый и эффективный способ для доступа к самым важным командам редактора **Writer** - это кнопки на панелях инструментов, которые располагаются ниже главного меню и дублируют его команды. С каждой кнопкой связана некоторая операция. Если вы задержите курсор мыши на какой-либо кнопке, то через несколько секунд рядом появится подсказка относительно функции данной кнопки. Просмотрите все кнопки.

Сразу после запуска редактора **Writer**, по умолчанию в его окне отображаются стандартная панель и панель форматирования. Всего имеется более десяти панелей инструментов, каждую из них можно вывести на экран или, наоборот - спрятать.

Чтобы узнать какие еще панели имеются в «запаснике» редактора **Writer**, выберете команду **Панели инструментов** в меню **Вид**. В открывшемся окне вложенного меню появится список всех существующих панелей. Те из них, что в настоящий момент присутствуют на экране отмечены флажком в виде галочки рядом с названием панели. Для вызова нужной панели щелкните мышью в строке ее названия, и панель отобразится на экране. Аналогичным образом убрать панель.

### **Добавление кнопок на панели инструментов.**

Чтобы добавить кнопку на панель инструментов, выполните следующие действия:

1. выберете команду **Сервис => Настройка**.
2. выберете вкладку **Панели инструментов**
3. выберете саму панель инструментов, в которой хотите добавить команду.
4. при выборе элемента в списке **Команды** отображаются команды, относящиеся к выбранной категории.
5. отыскав команду, которую вы хотите добавить на панель инструментов, щелкните на окошке рядом с командой, появиться галочка, означающая добавление команды

6. щелкните на кнопке **Закрыть**, чтобы выйти из диалогового окна **Настройка**

### **Удаление кнопок панелей инструментов**

Откройте диалоговое окно **Настройка**, выбрав команду **Сервис => Настройка**. Активизация диалогового окна **Настройка** позволит удалить любую кнопку, сняв галочку с команды которую хотите удалить. Щелкните на кнопке **Закрыть** для запоминания изменений и закрытия диалогового окна **Настройка**.

### **Контекстное меню.**

В отдельных случаях для ускорения работы имеет смысл вообще отказаться от пользования главным меню и панелями инструментов. Нужную команду можно вызвать щелчком правой кнопки мыши. При этом появится **контекстное меню**, которое видоизменяется в зависимости от объекта щелчка.

### **Полосы прокрутки.**

Справа и внизу окна документа располагаются полосы вертикальной и горизонтальной прокрутки текста. Они применяются в тех случаях, когда весь текст вашего документа не умещается на экране и требует прокрутки вверх-вниз либо вправо-влево. Это достигается:

- Щелчком мыши на стрелках в полосах прокрутки.
- Щелчком мыши на полосе прокрутки с соответствующей стороны бегунка.
- Используя клавиши управления курсором.
- **PageUp** - на экран вверх.
- **PageDown** - на экран вниз.
- **Ctrl+Home** - к началу документа.
- **Ctrl+End** - в конец документа.

Для быстрого перемещения по тексту очень удобно использовать бегунок, перетаскивая его мышью вдоль полосы прокрутки.

#### **Масштабные линейки.**

Масштабные линейки располагаются сверху и слева от вашего документа. С их помощью вы можете изменять абзацный отступ, масштаб отображения страниц и текста на экране, ширину колонок текста и размеры ячеек таблиц. Включение - выключение линеек достигается выбором команды Линейки в меню Вид.

#### **СОЗДАНИЕ ДОКУМЕНТА В РЕДАКТОРЕ Writer**

Чтобы создать новый документ **Writer**, можно либо выбрать команду Файл => Создать, либо щелкнуть на кнопке Создать, расположенной на панели инструментов Стандартная. В первом случае появится диалоговое окно Создание документа. Выберите в нем стандартный шаблон Новый документ и щелкните на кнопке ОК. Лучше всего пользоваться кнопкой на панели инструментов.

В открывшемся рабочем поле окна документа **Writer** вы увидите мерцающий курсор, а в строке заголовка **Writer** - надпись Безымянный 1. Таким образом, **Writer** по умолчанию присвоил документу стандартное имя Безымянный 1, следующий будет назван Безымянный 2 и т.д.

#### **Сохранение документа.**

Для сохранения документа ему необходимо присвоить имя чтобы потом можно было найти в папке среди других. Для этого поступите следующим образом:

- Щелкните на кнопке **Сохранить** или выберите команду **Сохранить** из меню **Файл**. На экране появится диалоговое окно **Сохранение документа**.

- В поле **Имя файла** введите название.

- Щелкните на кнопке **Сохранить**.

Ваш документ сохранится в файле с названным вами именем и с расширением **\*.odt**.

#### **Сохранение файла под другим именем.**

Если вам по какой-либо причине захотелось сохранить документ под новым именем, выберите команду **Сохранить как** в меню **Файл**. Введите новое имя и щелкните на кнопке **Сохранить**.

#### **Повторное сохранение файла.**

Сохранив документ однажды, вы сможете сохранить его в следующий раз, уже не вызывая для этого диалоговое окно Сохранение документа. Для этого щелкните на кнопке **Сохранить** на стандартной панели инструментов.

#### **Открытие сохраненных файлов.**

Для того, чтобы открыть файл можно щелкнуть на кнопке **Открыть** стандартной панели инструментов или выбрать аналогичную команду из меню **Файл**. Откроется диалоговое окно.

**Открытие документа.** Теперь нужно, выбрав соответствующий диск, выделить имя файла в раскрывшемся списке и дважды щелкнуть на нем либо щелкнуть на кнопке **Открыть**.

Если у вас открыто несколько документов, то переключение между открытыми документами производится выбором нужного документа из меню **Окно**.

### 3 Наименование вопроса № 3

Write – текстовый процессор.

**Текстовый процессор** — компьютерная программа, используемая для написания и модификации документов, компоновки макета текста и предварительного просмотра документов в том виде, в котором они будут напечатаны (свойство, известное как WYSIWYG).

Современные текстовые процессоры, помимо форматирования шрифтов и абзацев и проверки орфографии, включают возможности, ранее присущие лишь настольным издательским системам, в том числе создание таблиц и вставку графических изображений.


Наиболее известными примерами текстовых процессоров являются Microsoft Word и OpenOffice.org Writer.

Текстовыми процессорами в 1970-е — 1980-е годы называли предназначенные для набора и печати текстов машины индивидуального и офисного использования, состоящие из клавиатуры, встроенного компьютера для простейшего редактирования текста, а также электрического печатного устройства. Позднее наименование «текстовый процессор» стало использоваться для компьютерных программ, предназначенных для аналогичного использования.

Программы для работы с текстами можно разделить на текстовые редакторы, текстовые процессоры и издательские системы.

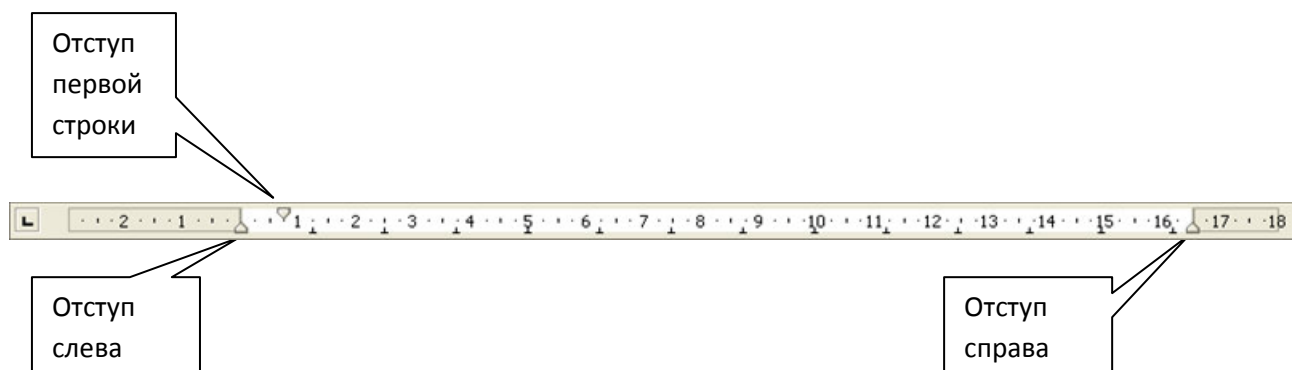
### 4 Наименование вопроса № 4

Параметры абзаца

Формально слово «абзац» означает отступ новой строки в тексте. Как понятие русского языка абзац – это законченная по смыслу часть литературного произведения. В текстовом редакторе абзац – последовательность символов, заканчивающаяся символом (маркером) конца абзаца, который вводится нажатием клавиши **<Enter>**. Если включить режим показа непечатаемых знаков, то конец каждого абзаца будет помечен таким символом:  .

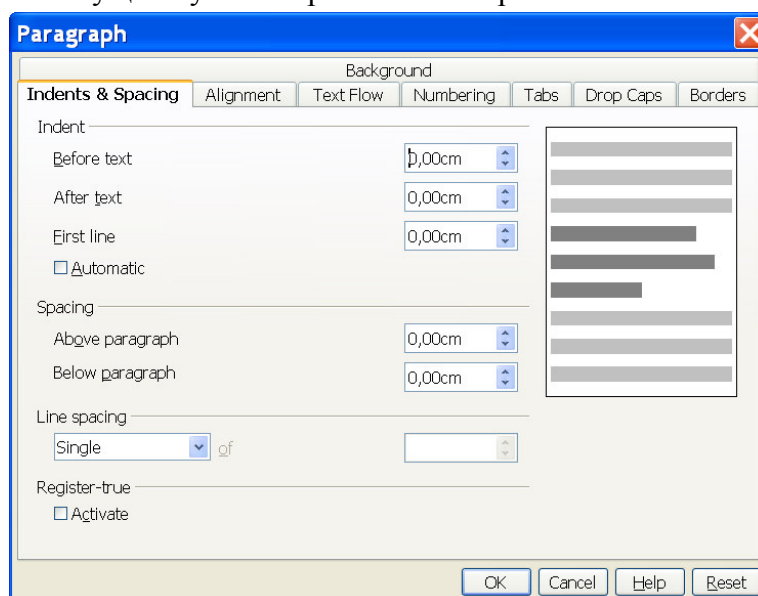
Ширина текста на странице определяется положением кнопок «Отступ слева» (Before text), «Отступ справа» (After text), «Отступ первой строки» (First line), расположенных на Горизонтальной Линейке. Передвигая кнопки отступов, можно задать ширину абзаца на ширине страницы. Эти действия можно выполнять как до начала ввода текста, так и после. Отступы будут применяться ко всему вводимому тексту до тех пор, пока установки не будут изменены.





Используя диалоговое окно, вызываемое пунктом меню **Формат(Format) → Абзац(Paragraph...)** для абзаца можно задать интервалы перед (Spacing Above Paragraph) и после абзаца (Spacing Below Paragraph), межстрочный интервал (Line spacing), а также уже знакомые параметры ширины текста на странице «Отступ слева» (Before text), «Отступ справа» (After text), «Отступ первой строки» (First line), - расстояния в сантиметрах, измеряемые от поля страницы до текста.

Выравнивание абзацев (Alignment) – это расположение текста на печатной странице относительно ее краев. Чаще всего это горизонтальное выравнивание текста, то есть расположение его между левым и правым полями страницы, но можно также задать и вертикальное выравнивание. Существует четыре способа выравнивания текста:

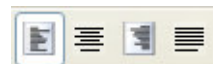


- При выравнивании по левому краю (Align left) все строки абзаца начинаются с одной и той же позиции. Заканчивается каждая строка в своей позиции. Такое выравнивание является стандартом англоязычного документа.

- При выравнивании по правому краю (Align right) строка может начинаться в любой позиции, но заканчиваются все они в крайней правой позиции. Так можно выравнивать текст той части бланка заявления, где пишется обращение «кому», «от кого».

- При выравнивании по центру (Centered) текст симметричен относительно середины страницы. Так выравнивают заголовки текстов.

- При выравнивании по ширине (Justified) оба края совершенно ровные. Это достигается за счет увеличения количества пробелов между словами. Так оформляют документы на русском языке и в печатных изданиях.



Кнопки выравнивания расположены на панели инструментов.

Назначение кнопки понятно по рисунку на ней, но можно также воспользоваться всплывающими подсказками, которые появляются, если подвести стрелку указателя мыши к кнопке на панели инструментов. Действие выравнивания распространяется на весь абзац.

## 1. 10 Лекция № 10 ( 2 часа).

Тема: «Base 2003»

### 1.10.1 Вопросы лекции:

1. Основные понятия баз данных
2. Система управления базами данных Base 2003 и ее основные возможности
3. Создание базы данных «Деканат»

### 1.10.2 Краткое содержание вопросов: *(тезисно изложить основное содержание рассматриваемых вопросов)*

1. Наименование вопроса № 1  
Основные понятия баз данных.

**База данных**- это набор сведений, относящихся к определенной теме или задаче: отслеживание заказов клиентов, хранение товаров на складе, сведения о сотрудниках учреждения и т.п.

Работа с большими наборами данных легче автоматизируется, когда данные упорядочены, то есть образуют заданную структуру. Существует три основных типа структур данных: **линейная или список**(например, листы книги), **иерархическая**(части, разделы, главы, параграфы книги) и **табличная**(оглавление или содержание книги).

**Линейные структуры данных (списки)**- это упорядоченные структуры, в которых адрес элемента однозначно определяется его номером.

**В иерархической структуре** адрес каждого элемента определяется путём доступа (маршрутом), ведущим от вершины структуры к данному элементу.

Основным недостатком иерархических структур данных является увеличенный размер пути доступа. Часто бывает, что длина маршрута оказывается больше, чем длина самих данных, к которым он ведёт.

**Табличные структуры данных (матрицы)**– это упорядоченные структуры, в которых адрес элемента определяется номером строки и номером столбца, на пересечении которых находится ячейка, содержащая искомый элемент,

**База данных (БД)** – это организованная структура, предназначенная для хранения информации (в современных базах данных хранятся не только данные, но и информация).

По тому, какой тип структуры данных реализован в базе данных, различают **иерархическую**, **сетевую** (усовершенствованная иерархическая) и **реляционную** (табличную) базы данных. Наибольшую популярность приобрели **реляционные базы данных**, т.к. они наиболее наглядны и удобны в работе. Однако недостатком их является дублирование данных. Уменьшения дублирования данных добиваются разумной организацией таблиц и установлением связей между ними.

Таблица БД имеет следующую структуру. Графы таблицы называются **полями**, причём каждое поле имеет **имя** и **тип**. Строки же таблицы называются **записями**.

2. Наименование вопроса № 2



## Система управления базами данных Base 2003 и ее основные возможности.

Представление информации в таблице - наилучший способ структурирования данных. Все данные записаны в клеточках таблицы по определенным правилам - форматам, одинаковым для всего столбца. Все столбцы имеют названия. Кроме этого нетрудно заметить, что фамилии студентов записаны по алфавиту, при этом для записей имен и фамилий используются заглавные буквы. Каждая строка таблицы имеет порядковый номер.

Автоматизировать обработку данных, которые хранятся в неструктурированном виде сложно, а порой и просто невозможно. Поэтому вырабатывают определенные соглашения о способах представления данных. Обычно это делает разработчик базы данных. В результате все реквизиты имеют одинаковый вид и тип данных, что делает их структурированными и позволяет создать базу данных. В результате можно сказать, что:

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

Обработывает структурированные данные централизованный программный механизм, который называется системой управления базами данных.

**Система управления базами данных (СУБД)** - это программный механизм, предназначенный для записи, поиска, сортировки, обработки (анализа) и печати информации, содержащейся в базе данных.

В компьютерной базе данных информация представляется в виде **таблицы**, очень похожей на электронную таблицу. Названия столбцов, представляющих «шапку» таблицы, называют **именами полей или реквизитами**, а сами столбцы - **полями**. Данные в полях называют **значениями реквизитов или значениями полей**. Для описания поля, кроме его имени используются следующие характеристики и свойства полей:

**Тип поля.** Подобно электронной таблице, работающей с тремя типами полей: текстовый, числовой и формула, в таблицах используется несколько большее количество типов полей.

**Длина поля** - максимально возможное количество символов.

**Точность** (для числовых типов полей) - количество знаков после запятой.

**Маска ввода** - форма средства автоматизации ввода, в которой вводятся данные в поле. Например, одно и то же значение имеют поля даты: 03.03.95 или 03.03.1995, или 03-март-1995, но отличаются по формату.

**Сообщение об ошибке** - текстовое сообщение, которое выдается в поле при попытке ввода ошибочных данных.

**Условие на значение** - ограничение, используемое для проверки правильности ввода данных.

**Пустое и обязательное поле** - свойство поля, определяющее обязательность заполнения поля при наполнении базы данных.

**Индексированное поле** - дополнительное имя поля, позволяющее ускорить операции поиска и сортировки записей.

Строки данных таблицы называются **записями** (рис.3).

Таким образом:

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

**Запись** - совокупность логически связанных полей, представленных одной строкой таблицы.

**Файл (таблица)** - совокупность экземпляров записей одной структуры.

Строки-записи    Столбцы=поля

Имена полей

Фамилия	Имя	Отчество	Клички	
Авраменко	Юрий	Иванович	Авраам	
Белов	Виктор	Петрович	Белый	
Петрова	Галина	Сергеевна	Худоба	
Сидоров	Иван	Захарович	Захар	

Рис. 1. Основные структурные элементы базы данных

Классификация баз данных

Многообразие характеристик и видов баз данных порождает многообразие классификации. Рассмотрим основные виды классификации.

По технологии обработки данных базы данных подразделяются на централизованные и распределенные.

Централизованная база данных хранится в памяти одной вычислительной системы, к которой подключены несколько других компьютеров.

Распределенная база данных состоит из нескольких, возможно пересекающихся или даже дублирующих друг друга частей, хранимых в различных ПК компьютерной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

По способу доступа к данным базы данных подразделяются на базы данных с локальным доступом и базы данных с удаленным (сетевым) доступом.

Системы централизованных баз данных с сетевым доступом предполагают различные архитектуры подобных систем:

**файл-сервер.** Согласно этой архитектуре в компьютерной сети выделяется машина- сервер для хранения файлов централизованной базы данных. Файлы базы данных могут быть переданы на рабочие станции для обработки: ввода, корректировки, поиска записей. При большой интенсивности доступа к одним и тем же файлам производительность системы падает. В этой системе сервер и рабочие станции должны быть реализованы на достаточно мощных компьютерах.

**клиент-сервер** - архитектура, используемая не только для хранения файлов централизованной базы данных на сервере, но и выполняющая на том же сервере основной объем работы по обработке данных. Таким образом, при необходимости поиска информации в базе данных рабочим станциям - клиентам передаются не файлы данных, а уже записи, отобранные в результате обработки файлов данных. Такая архитектура позволяет использовать маломощные компьютеры в качестве рабочих станций, но обязательно в качестве сервера используется очень мощный компьютер.

Прежде чем создавать базу данных, с которой вам придется работать, необходимо выбрать модель данных, наиболее удобную для решения поставленной задачи.

### 3 Наименование вопроса № 3

Создание базы данных «Деканат»

Описание структуры базы данных Структура данных – это организационная схема данных, в соответствии с которой они упорядочены, с тем, чтобы их можно было интерпретировать и выполнять над ними определенные операции. База данных «Деканат ВУЗа» является реляционной. В процессе ее разработки были созданы следующие

таблицы: Kafedra (Кафедра); Gruppa (Группы); Students (Студенты); Distciplina (Дисциплина); Sotrudniki (Сотрудники); Raspisanie (Расписание); Results (Результаты); Result\_otsenka (Результат\_оценка); Otsenka (Оценка); Oplata (Оплата). Также в базе данных имеются хранимые процедуры, реализованные на языке T-SQL. Также в базе данных определены следующие хранимые процедуры: Dbo.pr1 Dbo.pr2 3.2 Описание свойств таблиц БД Таблицы в БД – это объекты базы данных, предназначенные для хранения пользовательских данных.

### **1.11 Лекция № 11 ( 4 часа).**

**Тема:** «Классификация компьютерных сетей. Интернет как единая система ресурсов»

#### **1.11.1 Вопросы лекции:**

1. Введение
2. Топология сети
3. Электронная документация: определение и особенности
4. Типы Интернет-приложений.

**1.11.2 Краткое содержание вопросов:** *(тезисно изложить основное содержание рассматриваемых вопросов)*

1. Наименование вопроса № 1  
Введение.

Сетевая инфраструктура играет весьма важную роль в обеспечении безопасности web-сервера. В большинстве конфигураций сетевая инфраструктура является первой линией обороны между Интернетом и web-сервером. Хотя на создание той или иной сетевой инфраструктуры влияют многие факторы, отличные от проблем, связанных с безопасностью (такие как цена, производительность и надежность), мы в основном будем рассматривать проблемы, относящиеся к безопасности.

Сеть, которая разрабатывалась без учета требований обеспечения безопасности web-сервера, может иметь серьезные уязвимости для его безопасности. Частота, изощренность и разнообразие web-атак, осуществляемых сегодня, говорят о том, что безопасность web должна обеспечиваться с помощью многоуровневых и распределенных механизмов обороны (так называемая "оборона вглубь"). Обсудим сетевые компоненты, которые могут поддерживать и защищать web-серверы с целью соответствующего усиления безопасности.

2. Наименование вопроса № 2  
Топология сети.

Существует много вариантов создания топологии сети, и безопасность может быть не главным фактором при принятии решений. Топология сети является первым и во многих аспектах наиболее критичным сетевым решением, которое воздействует на безопасность web-сервера. Она важна по нескольким причинам. Топология сети определяет, какая сетевая инфраструктура может быть использована для защиты web-сервера. Например, если web-сервер расположен позади firewall'a организации, то firewall может использоваться для контроля входящего и исходящего трафика во внутреннюю сеть и web-сервер. Топология сети также определяет, какие участки сети окажутся уязвимыми, если web-сервер будет скомпрометирован. Например, если web-сервер размещен во внутренней производственной сети, то внутренняя сеть может быть подвергнута атаке со стороны скомпрометированного web-сервера. Организация может

принять решение не размещать web-сервер в своей сети, а разместить его в сторонней организации.

Иногда принимается решение о размещении публичных web-серверов во внутренних производственных сетях, где расположены их внутренние серверы и рабочие станции пользователей. Такое расположение не рекомендуется, потому что при этом существует неоправданный риск компрометации внутренней сети. Принципиально слабым местом такой конфигурации является то, что web-серверы часто являются целью атакующих. Если удастся скомпрометировать web-сервер, атакующие окажутся во внутренней сети и могут легко скомпрометировать внутренние хосты.

Также не рекомендуется располагать web-сервер перед firewall'ом организации или роутером, который осуществляет IP-фильтрацию. При таком типе конфигурации сеть может обеспечивать меньшую, чем необходимо, защиту web-сервера. Вся безопасность зависит только от самого web-сервера, который представляет собой единственную точку падения. Даже если при такой топологии и задействовать существенные меры обеспечения безопасности, все равно ОС, на которой выполняется web-сервер, должна быть достаточно укрепленной (все ненужные и небезопасные сервисы запрещены) и полностью применены необходимые patches безопасности, а web-администратор должен выполнять своевременное устранение всех уязвимостей. Другой недостаток подобной топологии состоит в том, что в данном случае трудно реализовать какой-либо тип безопасного удаленного администрирования или модификации содержимого.

### 3. Наименование вопроса № 3

Электронная документация: определение и особенности

Документ является основным способом представления информации. Электронный документ — это бумажный документ, введенный в компьютер для обработки. Финансовые электронные документы могут снабжаться электронной подписью. Электронные документы бывают структурированными, и тогда они находятся в базах данных, и неструктурированными, содержащими тексты на естественном языке.

В общем же принято считать под электронным документом (ЭД) структурированный информационный объект, в соответствие которому может быть поставлена совокупность файлов, хранящихся на накопителе компьютера. Необходимым признаком ЭД является "регистрационная карточка", состоящая из реквизитов документа, содержащих перечень необходимых данных о нём. Согласно законодательству, электронным является документ, в котором информация представлена в электронно-цифровой форме.

Электронные документы можно разделить на два основных типа: неформализованные (произвольные) и служебные (официальные). Неформализованный электронный документ — это любое сообщение, записка, текст, записанный на машинном носителе. Под служебным электронным документом понимается записанное на машинном носителе электронное сообщение, реквизиты которого оформлены в соответствии с нормативными требованиями.

Электронные документы по сравнению с бумажными обладают следующими преимуществами:

- более низкая стоимость и время передачи электронного документа из одного места в другое;
- более низкая стоимость и время тиражирования ЭД;
- более низкая стоимость архивного хранения ЭД;
- возможность контекстного поиска;
- новые возможности защиты документов;
- упрощение подготовки ЭД в сочетании с широкими возможностями;

- принципиально новые возможности представления ЭД. Документ может иметь динамическое содержание (например, аудио-, видеовставки).

### **Электронный документооборот**

Основные принципы построения системы электронного документооборота:

- соответствие требованиям стандартов на формы и системы документации;
- распределённость;
- масштабируемость;
- модульность;
- открытость;
- переносимость на другие аппаратные платформы.

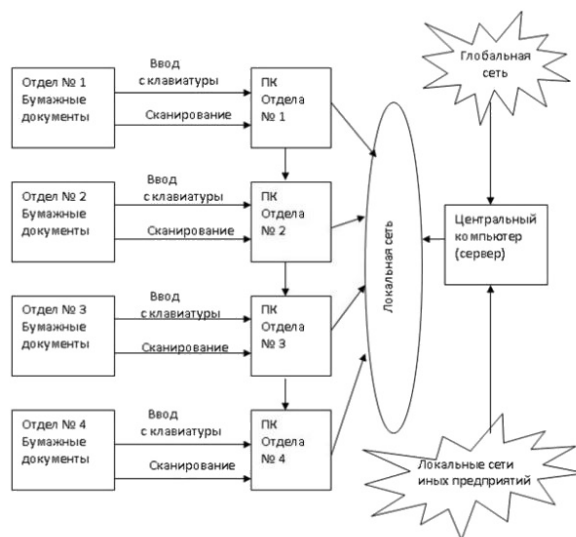
Основными функциями системы электронного документооборота являются:

- регистрация документов;
- контроль исполнения документов;
- создание справочников и работа с ними;
- контроль движения бумажного и электронного документа, ведение истории работы с документами;
- создание и редактирование реквизитов документов;
- формирование отчетов по документообороту предприятия;
- импорт документов из файловой системы и Интернета;
- создание документа прямо из системы на основе шаблона (прямая интеграция);
- работа с версиями документа, сложными многокомпонентными и многоформатными документами, вложениями;
- электронное распространение документов;
- работа с документами в папках;
- получение документов посредством сканирования и распознавания.
- уменьшением затрат на доступ к информации и обработку документов.

Системы документооборота обычно внедряются, чтобы решать определённые задачи, стоящие перед организацией. К таким задачам можно отнести следующие:

- обеспечение более эффективного управления за счёт автоматического контроля выполнения, прозрачности деятельности всей организации на всех уровнях;
- поддержка эффективного накопления, управления и доступа к информации и знаниям;
- исключение бумажных документов из внутреннего оборота предприятия;
- исключение необходимости или существенное упрощение и удешевление хранения бумажных документов за счёт наличия оперативного электронного архива;
- экономия ресурсов за счёт сокращения издержек на управление потоками документов в организации;
- поддержка системы контроля качества, соответствующей международным нормам;
- обеспечение кадровой гибкости за счёт большей формализации деятельности каждого сотрудника и возможности хранения всей предыстории его деятельности;
- протоколирование деятельности предприятия в целом (внутренние служебные расследования, анализ деятельности подразделений, выявление "горячих точек" в деятельности);
- оптимизация бизнес - процессов и автоматизация механизма их выполнения и контроля.

Электронные потоки информации на предприятии (в офисе) схематично представлены на [рис.1](#).



**Рис. 1.** Общая схема электронного документооборота

Документооборот — это прохождение документов по соответствующим отделам до сдачи их в архив. Электронный документ может появиться либо в результате ввода бумажного документа с помощью клавиатуры или сканера, либо из глобальной сети, либо из локальной сети сторонних организаций.

Всякая система электронного документооборота должна быть распределённой, так как ввод документов происходит на территориально-распределённой организации.

Перечислим функции систем электронного документооборота.

1. Хранение электронных документов. Носители электронных документов характеризуются двумя параметрами:

1. стоимостью хранения 1 мегабайта информации;
2. скоростью доступа к информации. Задача заключается в выборе оптимального носителя.

2. Организация поиска документов. Существуют два типа поиска. Во-первых, атрибутивный поиск, когда каждому документу присваивается набор идентифицирующих его атрибутов. Поиск документа осуществляется путём сравнения значений этих атрибутов со значениями запроса. Примерами атрибутов могут служить код поставщика, код или фамилия служащего, время создания документа и т.д.

Во-вторых, полнотекстовый поиск, когда документ отыскивается по словам, входящим в сам документ.

Для поиска известного документа используется атрибутивный тип, для неизвестного — полнотекстовый.

3. Поддержка защиты документов от несанкционированного доступа. Каждый документ должен иметь список пользователей, имеющих право доступа к нему.

4. Маршрутизация документов. Документы различаются по типам носителей информации. Существуют три типа носителей: 1) бумага; 2) микрофиши: с точки зрения поиска мало чем отличаются от бумаги, но занимают меньше места; 3) электронные носители — жёсткие, оптические, магнитооптические диски.

От момента первой записи до сдачи в архив документы проходят различные преобразования ([рис.2](#)).

Документы могут существовать одновременно в нескольких видах, переходя из одного вида в другой. В каком виде должен храниться документ, зависит от ограничений, накладываемых пользователем на стоимость, и время поиска документа, а также от стоимости передачи документа от одного рабочего места к другому.

Массовый ввод документов, которые, как правило, не подлежат арифметической обработке (тексты), осуществляется с помощью операции сканирования. Для этого вначале выполняется подготовка к вводу документа, а затем ввод.



**Рис. 2.** Схема преобразования различных видов документов

Подготовка состоит в следующем: если документ новый, то он описывается и регистрируется как новый класс документа. Для этого документ сканируется (незаполненный) и создаётся форма по отсканированному шаблону. Далее определяются те поля, которые будут распознаваться системой или заполняться пользователем с клавиатуры. Указываются типы данных обрабатываемых полей. После сканирования документы автоматически направляются на операцию распознавания. Но перед этим система осуществляет ряд операций, улучшающих изображение символов (выравнивание, удаление шума, восстановление символов и др.). Особенно это важно для рукописных данных.

Одна из важных функций системы документооборота — маршрутизация и контроль исполнения. Маршрутизация сообщений в системе электронного документооборота — это построение схемы, согласно которой они передаются между рабочими местами пользователей АРМ.

Известны следующие системы маршрутизации:

1. Свободная маршрутизация — последовательная или параллельная. При последовательной маршрутизации документ проходит от одного пользователя к другому, а при параллельной он одновременно поступает к нескольким пользователям.

2. Свободная маршрутизация с контролем исполнения. Под контролем понимается: контроль доставки документа; контроль исполнения (выдача извещения, что задание выполнено); мониторинг задания (кто и что сейчас делает с заданием).

3. Маршрутизация по заранее определённым маршрутам с контролем исполнения.

4. Система электронной почты.

В сети Интернет электронный документооборот организуется в той группе сервисов, которая обеспечивает поиск информации с помощью Web-серверов.

Все Web-серверы делятся на две группы: 1) серверы управления трафиком (traffic); 2) конечные серверы.

Пользователь с помощью серверов первой группы отыскивает необходимую информацию, а потом обращается за помощью серверов второй группы. Наиболее мощные средства в группе серверов управления трафиком — это порталы, т.е. Web-узлы, представляющие собой сочетание базовых услуг, например, информационного поиска и передачи найденной информации с помощью электронной почты. Главное отличительное свойство портала в том, что он не только отыскивает для пользователя необходимую информацию, но ещё и обеспечивает набор сервисных услуг.



Конечный сервер может содержать рекламный сервер, состоящий из 1—2 страниц, информационный сервер, Интернет - магазин, Интернет - витрину.

Чтобы быть доступными для конечного сервера, электронные документы описываются с помощью специального гипертекстового языка HTML, представляющего собой набор правил для описания структуры любого электронного документа.

#### 4. Наименование вопроса № 4

##### Типы Интернет-приложений

Web-приложение это web-система, позволяющая пользователям реализовать доступ к бизнес-логике через браузер. Web-система это система гипермедиа, поскольку ее ресурсы связаны между собой. Термин "web" означает, что система рассматривается как набор узлов с перекрестными ссылками.

##### **Типы Интернет-приложений**

Существует четыре типа Интернет-приложений:

- Web-приложения, которые работают на сервере, передавая через Интернет данные на клиентские машины. Для их применения требуются Web-браузеры, такие, как Microsoft Internet Explorer и Netscape Navigator;
- Web-сервисы, которые позволяют приложениям обрабатывать их данные на сервере. При этом передача подлежащих обработке данных на сервер и возврат результатов осуществляется через Интернет;
- приложения с поддержкой Интернета автономные программы со встроенными механизмами, позволяющими их пользователям регистрироваться, получать обновления, а также предоставляющими доступ к справочной системе и другим вспомогательным службам через Интернет;
- одноранговые приложения автономные программы, использующие Интернет для взаимодействия с другими программными продуктами этого же типа.

##### **Архитектурные шаблоны Интернет-приложений**

Под архитектурой системы понимается высокоуровневое представление архитектурно-значимых компонентов системы. В этом смысле компонент представляет собой отдельную сущность с открытым интерфейсом. Архитектурно-значимые компоненты это те, которые входят в представление системы на самом высоком уровне. Обычно эти компоненты или их набор нельзя сгруппировать или объединить в пакет с другими аналогичными компонентами. Свойства отдельных компонентов будут зависеть от особенностей представления системы.

Существует широкий спектр продуктов и технологий, которые могут быть частью архитектуры web-приложений. Архитектурный шаблон отражает фундаментальную структурно-организационную схему программных систем. Он представляет набор предопределенных подсистем, описывает спектр их обязанностей, а также определяет правила и рекомендации для организации взаимодействия между ними.

Можно выделить следующие архитектурные шаблоны web-приложений:

1. Шаблон Thin Web Client (на основе "тонкого" Web-клиента) используется в большинстве приложений Internet и предоставляет ограниченные возможности по управлению конфигурацией клиента. В распоряжении клиента должен быть только стандартный браузер, поддерживающий формы. Все операции, связанные с бизнес-логикой, выполняются на сервере. Этот шаблон больше всего подходит для Web-приложений, в которых клиент обладает минимальными вычислительными возможностями или не может управлять своей конфигурацией.

2. Шаблон Thick Web Client (на основе "толстого" Web-клиента) предполагает, что значительная часть бизнес-логики выполняется на клиентской машине. Обычно для



выполнения бизнес-логики клиентом используется DHTML, апплеты Java или управляющие элементы ActiveX. Взаимодействие с сервером также происходит через протокол HTTP.

Шаблон Web Delivery (на основе механизма Web-доставки). При взаимодействии клиента и сервера, кроме протокола HTTP, используются и другие протоколы, такие как IOOP (Internet Inter-Orb Protocol) и DCOM, которые могут применяться для поддержки системы распределенных объектов. В данном случае браузер функционирует как контейнерный модуль системы распределенных объектов.

### **Виды web-серверов**

Различают статические и активные серверы Web. Если страницы сервера содержат только статическую текстовую и мультимедийную информацию, а также гипертекстовые ссылки на другие страницы, то сервер называется статическим. Если страницы web-сервера изменяют своё содержимое в зависимости от действий пользователя, то такие серверы называют активными. Статический сервер Web не может служить основой для создания интерактивных приложений с доступом через Интернет, так как он не предусматривает никаких средств ввода и обработки запросов.

## **1. 12 Лекция № 12 ( 6 часов).**

**Тема:** «Создание почтового ящика»

### **1. 12. 1 Вопросы лекции:**

- 1.Электронная почта.
- 2.Телеконференции

**1.12.2 Краткое содержание вопросов:** *(тезисно изложить основное содержание рассматриваемых вопросов)*

- 1.Наименование вопроса № 1
- Электронная почта

### ***Электронная почта.***

Электронная почта (e-mail) - наиболее распространенный сервис Интернета, так как она является исторически первой информационной услугой компьютерных сетей и не требует обязательного наличия высокоскоростных и качественных линий связи.

Широкую популярность электронная почта завоевала потому, что имеет несколько серьезных преимуществ перед обычной почтой. Наиболее важное из них - это скорость пересылки сообщений. Если письмо по обычной почте может идти до адресата дни и недели, то письмо, посланное по электронной почте, сокращает время передачи до нескольких десятков секунд или, в худшем случае, до нескольких часов.

Другое преимущество состоит в том, что электронное письмо может содержать не только текстовое сообщение, но и вложенные файлы (программы, графику, звук и пр.). Однако не рекомендуется пересылать по почте слишком большие файлы, так как это замедляет работу сети. Для того чтобы этого не происходило, на некоторых почтовых серверах вводятся ограничения на размер пересылаемых сообщений (обычно почтовый сервер не пропускает сообщения более 2 000 000 байтов).

Кроме того, электронная почта позволяет:

- посылать сообщение сразу нескольким абонентам;
- пересылать письма на другие адреса;
- включить автоответчик, на все входящие письма будет автоматически

отсылаться ответ;

- создать правила для выполнения определенных действий с однотипными сообщениями (например, удалять рекламные сообщения, приходящие от определенных адресов) и так далее.

**Адрес электронной почты.** Для того чтобы электронное письмо дошло до адресата, оно, кроме самого *сообщения*, обязательно должно содержать *адрес электронной почты* получателя письма.

Первая часть почтового адреса (*user\_name* - имя пользователя) имеет произвольный характер и задается самим пользователем при регистрации почтового ящика. Вторая часть (*server\_name* - имя сервера) является доменным именем почтового сервера, на котором пользователь зарегистрировал свой почтовый ящик.

*Адрес электронной почты записывается по определенной форме и состоит из двух частей, разделенных символом @: user\_name@server\_name*

Адрес электронной почты записывается только латинскими буквами и не должен содержать пробелов. Например, почтовый сервер компании МТУ-Интел имеет имя mtu-net.ru. Соответственно имена почтовых ящиков пользователей будут иметь вид:

user\_name@mtu-net.ru

**Функционирование электронной почты.** Любой пользователь Интернета может зарегистрировать почтовый ящик на одном из серверов Интернета (обычно на почтовом сервере провайдера), в котором будут накапливаться передаваемые и получаемые электронные письма. В настоящее время достаточно большое количество серверов Интернета предоставляют возможность бесплатно зарегистрировать почтовый ящик.

Для работы с электронной почтой необходимы специальные почтовые программы, причем для любой компьютерной платформы существует большое количество почтовых программ. Почтовые программы входят в состав широко распространенных коммуникационных пакетов: Outlook Express входит в Microsoft Internet Explorer, Netscape Messenger - в Netscape Communicator.

С помощью почтовой программы создается почтовое сообщение на локальном компьютере. На этом этапе кроме написания текста сообщения необходимо указать адрес получателя сообщения, тему сообщения и вложить в сообщение при необходимости файлы.

Процесс передачи сообщения начинается с подключения к Интернету и доставки сообщения в свой почтовый ящик на удаленном почтовом сервере. Почтовый сервер сразу же отправит это сообщение через систему почтовых серверов Интернета на почтовый сервер получателя в его почтовый ящик.

Адресат для получения письма должен соединиться с Интернетом и доставить почту из своего почтового ящика на удаленном почтовом сервере на свой локальный компьютер (рис. 4.9).

Почтовые программы обычно предоставляют пользователю также многочисленные дополнительные сервисы по работе с почтой (выбор адресов из адресной книги, автоматическую рассылку сообщений по указанным адресам и др.).



Рис. 4.9. Функционирование электронной почты

**Почтовая программа Outlook Express.** После запуска программы Outlook Express появится окно программы, которое состоит из четырех частей (рис. 4.10). В левой верхней части окна находится перечень папок, в которых хранится корреспонденция:

- *Входящие* - содержит получаемые адресатом письма;
- *Исходящие* - содержит отправляемые адресатом письма с момента их создания и до момента их доставки с локального компьютера пользователя на почтовый сервер провайдера;
- *Отправленные* - содержит все письма, доставленные на почтовый сервер;
- *Удаленные* - содержит удаленные письма;
- *Черновики* - содержит заготовки писем.

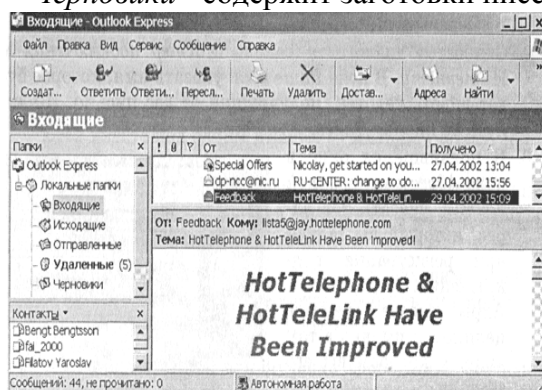


Рис. 4.10. Окно почтовой программы Outlook Express

Пользователь может создавать собственные папки для хранения тематически сгруппированных сообщений. В папках могут храниться не только сообщения, но и файлы, созданные с помощью других приложений.

В нижней левой части окна размещается *список контактов*, который предоставляет доступ к информации, хранящейся в *Адресной книге* (адреса электронной почты, телефоны и так далее).

Правое окно разделено на две части. В верхней части высвечивается список сообщений, хранящихся в выделенной папке.

В нижней части правого окна отображается содержание выделенного сообщения.

В первую очередь необходимо в соответствии с полученными в процессе регистрации почтового ящика данными (имя почтового ящика, пароль и др.) настроить почтовую программу. Создадим в почтовой программе Outlook Express учетную запись "Почта Интернета", при помощи которой можно будет отправлять и принимать электронную почту с конкретного почтового ящика.

### Создание учетной записи

1. В окне программы Outlook Express ввести команду [Сервис-Учетные записи]. Откроется диалоговая панель *Учетные записи Интернета*.

Выбрать вкладку *Почта*. Щелкнуть по кнопке *Добавить* и выбрать пункт *Почта...*

2. Откроется диалоговая панель *Мастер подключения к Интернету*. В поле *Ваше имя*: указать имя, которое будет видеть человек, получивший от вас письмо. Щелкнуть по кнопке *Далее*.

3. В появившемся окне поле *Адрес электронной почты*: указать тот адрес, который вы задали при регистрации подключения.

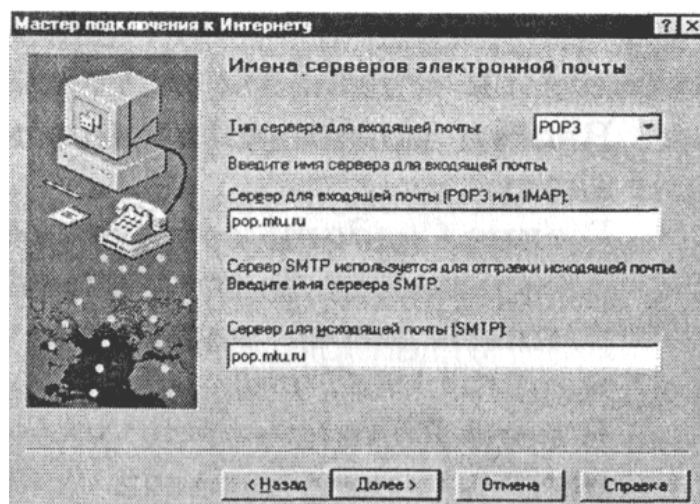
Адрес следует указать целиком и именно в том виде, в котором вы его создали.

Щелкнуть по кнопке *Далее*.



4. На появившейся диалоговой панели в поле *Тип сервера для входящей почты*: выберите POP3. Этот протокол наиболее часто используется для электронной почты.

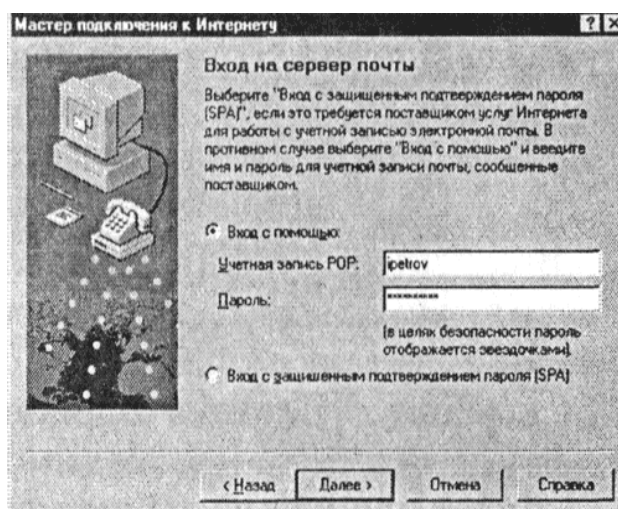
В полях *Сервер для входящей почты (POP3 или IMAP)*: и *Сервер для исходящей почты (SMTP)*: необходимо указать имена серверов входящей и исходящей почты, которые сообщает провайдер при регистрации подключения. Щелкнуть по кнопке *Далее*.



Теперь необходимо указать имя почтового ящика и пароль для входа на почтовый сервер.

5. В поле *Учетная запись POP*: ввести имя, которое вы указали при создании своего почтового адреса перед значком @.

В поле *Пароль*: необходимо указать тот пароль, который был получен при регистрации подключения у провайдера.



6. Заданные выше параметры электронной почты объединяются вместе под одним именем - именем учетной записи.

В поле *Имя учетной записи почты сети Интернет*: необходимо ввести имя для созданной учетной записи, например "Почта Интернета".

7. На следующих диалоговых панелях необходимо указать способ соединения с Интернетом, выбрать тип модема и используемое соединение с Интернетом.

**Создание, отправка и получение сообщений.** Создадим пробное сообщение в определенной кодировке с вложенным файлом.

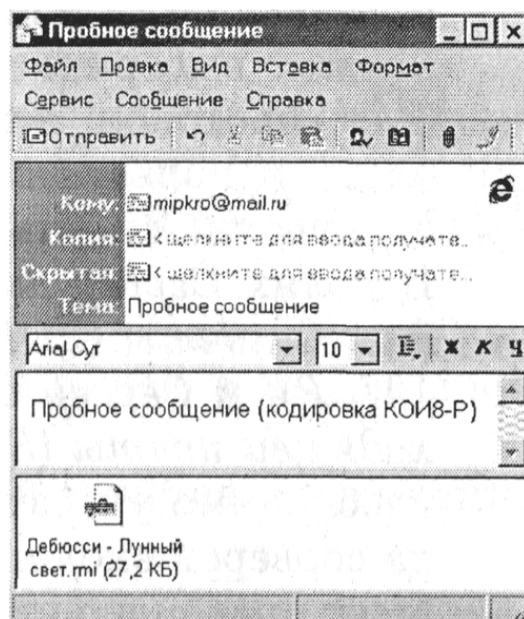
### Создание, отправка и получение сообщения

1. Ввести команду [Сообщение-Создать].

В окне *Создать сообщение* в поле *Кому*: необходимо указать электронный адрес адресата, например: `mipkro@mail.ru`

В поле *Копии*: можно указать адреса получателей копии сообщения.

В поле *Тема*: указывается тема сообщения, например "Пробное сообщение".



2. В области, отведенной для сообщения, вводится текст сообщения, например "Пробное сообщение (кодировка КОИ8-Р)".

Достаточно важен выбор правильной кодировки русских букв сообщения. При использовании электронной почтой чаще всего используются кодировки Windows и КОИ8-Р.

3. Выбор кодировки осуществить с помощью команды [Формат-Вид кодировки-Кириллица (КОИ8-Р)].

В сообщение можно вставлять файлы (текстовые, графические, звуковые и так далее).

4. Для вставки файла в сообщение необходимо ввести команду [Вставка-Вложение файла...]. В появившемся окне *Вставка* необходимо выбрать требуемый файл, и он будет вложен в сообщение.

5. Вставим, например, в сообщение звуковой файл Дебюсси-Лунный свет.ггтп из папки Media, которая находится в папке Windows. Название вложенного файла появится в нижней части окна сообщения.

Если создание сообщения производилось в автономном режиме без подключения к Интернету, сообщение необходимо сохранить в папке *Исходящие*.

6. После завершения работы над сообщением щелкнуть по кнопке *Отправить*, сообщение будет помещено в папку *Исходящие*.

Для того чтобы отправить сообщение адресату, необходимо подключиться к Интернету.

7. Щелкнуть по кнопке *Доставить почту*. Произойдет соединение с почтовым сервером, и все сообщения, находящиеся в папке *Исходящие* на локальном компьютере, будут доставлены на почтовый сервер. Одновременно отправленные сообщения будут перемещены на локальном компьютере в папку *Отправленные*.

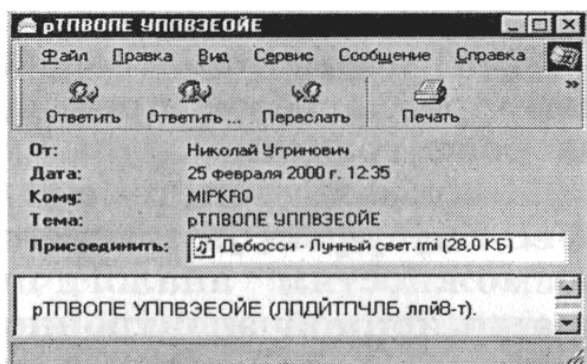
Почтовый сервер провайдера передаст сообщения в Интернет и через некоторое время они будут доставлены на почтовые сервера получателей. В данном случае пробное сообщение попадет в почтовый ящик [mirkro@mail.ru](mailto:mirkro@mail.ru)

Для получения сообщения абонент должен соединиться с Интернетом и произвести операцию доставки почты с почтового сервера провайдера на свой локальный компьютер.

8. Щелкнуть по кнопке *Доставить почту*. В процессе доставки почты сообщения, хранящиеся в почтовом ящике на почтовом сервере, будут переданы на локальный компьютер получателя и размещены в папке *Входящие*.

В случае установки кодировки, отличной от использованной при создании сообщения, сообщение будет представлять собой полную абракадабру.

9. В этом случае необходимо подобрать кодировку с помощью команды [Формат-Вид кодировки...].



### ***Электронная почта с Web-интерфейсом.***

В настоящее время для работы с электронной почтой используется Web-технология. В Интернете существуют Web-сайты, которые предлагают бесплатно зарегистрировать свой почтовый ящик всем желающим (например, <http://mail.ru>). В результате работа с почтой может производиться с помощью любого браузера после загрузки соответствующей Web-страницы, то есть специальные почтовые программы не требуются.

Для входа в такую почтовую систему зарегистрированным пользователям надо ввести свой идентификатор (логин) и пароль, а новым пройти процедуру регистрации.

Почтовая система с Web-интерфейсом предоставляет те же возможности, что и традиционная электронная почта. Отличием Web-почты является то, что все сообщения постоянно хранятся на удаленном сервере, а не на локальном компьютере пользователя.

Изобретатель электронной почты Рэй Томлинсон (Ray Tomlinson) официально признан разработчиком электронной почты для интернета. Его программа SNDMSG в 1971 году позволяла обмениваться почтой между разными компьютерами.

## **2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

Не предусмотрено РУП

## **3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ**

### **3.1 Практическое занятие №1 ( 2 часа).**

**Тема:** «Введение в информатику»

#### **3.1.1 Задание для работы:**

1. Что такое информация?
2. Какие действия человек совершает с информацией?
3. Как человек хранит информацию?
4. Носители информации.
5. Формы представления информации

#### **3.1.2 Краткое описание проводимого занятия:**

1. Что такое информация?

**Информация** - это сведения об окружающем мире, о происходящих в нем процессах и явлениях, воспринимаемые живыми организмами и техническими устройствами.

Человек получает информацию с помощью чувств:

Зрения;

Слуха;

Обоняния;

Вкуса;

Осязания.

## 2. Действия с информацией

*Действия с информацией* разнообразны:

1. Чтение газеты, книги (Рис. 1)



Рис. 1. Чтение газеты

2. Решение математических задач (Рис. 2)



Рис. 2. Решение задач

3. Фотографирование (Рис. 3)



Рис. 3. Фотографирование

4. Наблюдение (Рис. 4)





Рис. 4. Наблюдение

Действием с информацией не является приготовление обеда - это действие с продуктами.

Действия с информацией:

1. Получение;
2. Представление;
3. Передача;
4. Обработка;
5. Хранение;
6. Преобразование.

#### **4. Получение информации**

Наблюдение - это получение информации.

Смысл данного действия - воспринять информацию, получить сообщение.

#### **5. Представление информации**

Цель представления информации на носителе - ее хранение или передача.

Древние люди делали рисунки на скалах. В дальнейшем информацию передавали при помощи книг, картин. Современные способы передачи и хранения информации - магнитные ленты, дискеты, диски и др.

#### ***История носителей информации***

Для начала разберем, как хранили информацию в древнем мире.

*Наскальные изображения эпохи палеолита:*

Наскальная роспись (Рис. 5) - первый способ внешнего хранения информации в древние времена.



Рис. 5. Наскальная роспись

Художник этой эпохи, умевший изготавливать орудия своего труда и украшать стены пещер, по мнению ученых, должен уже был обладать и речью.

Наскальные рисунки показывали, как жили, охотились, выполняли ритуалы древние люди.

С возникновением человека разумного и появлением графического искусства развивалось устное общение.

*Изобретение письма:*

Впервые письменность появилась 5-4 тыс. до н.э. на территории современного Ирака, в двуречье Тигра и Евфрата.

Самые первые образцы письма (Рис. 6) - таблички из города Урук - небольшие глиняные бруски прямоугольной формы, на выпуклой поверхности которых нацарапаны пиктограммы.



Рис. 6. Образец древнего письма

В конце концов знаки приобрели форму клинообразных черточек, и это письмо получило название - клинопись.

При раскопках ассирийской столицы Ниневии были найдены таблички из царского хранилища, содержащие словари и грамматики шумерского, вавилонского и ассирийского языков.

*Папирус:*

Материалом, вытеснившим глиняные таблички и утвердившимся во всем античном мире после завоевания Египта Александром Македонским (332 год до н.э.), был папирус (Рис. 7).



Рис. 7. Папирус

Папирус был дорогим материалом, и писцы нередко использовали листы повторно, счищая старый текст.

*Заготовка папируса:*

Сначала папирус косили, затем разрезали каждый стебель на длинные полосы и накладывали их концами друг на друга. Рубцы (места стыков) сплющивали деревянным молотком. На этот кусок накладывали другой слой папируса, перпендикулярно первому, и снова били молотком, выделявшийся сок скреплял полосы. Чтобы получился длинный папирус, отдельные куски склеивали. Готовый папирус скатывали в свиток. Длина самого длинного из известных свитков папируса - 40,5 м.

### *Изобретение бумаги:*

В Древнем Китае, как и в Египте. Писали кистью и тушью на ткани (шелке), кости, черепаховом панцире, бамбуке. Шелк, однако, был дорог, поэтому постоянно велись поиски более экономичного материала. Изобретение бумаги (Рис. 8) приписывается Цай Луню, жившему во II веке н.э.



Рис. 8. Цай Лунь

### *Пергамент:*

Пергам (столица Мизии в Малой Азии) - один из крупнейших центров эллинского мира - знаменит тем, что согласно легенде, пергамский царь Евмений II изобрел пергамент.

Шкуры животных в качестве писчего материала использовались и раньше, но во II веке до н.э. Пергам становится одним из крупнейших производителей этого материала. Евмений основал знаменитую Пергамскую библиотеку, которая насчитывала более 200 000 свитков.

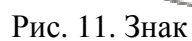
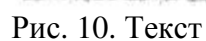


Рис. 9. Выделка пергамента

Смысл действия - отразить воспринятую информацию на носителе в удобной форме.

Представить информацию можно также различными способами:

1. Символами, знаками (Рис. 10 - 11);



**ШКАЛА ОТНОСИТЕЛЬНОЙ ЭЛЕКТРОПРИЦАТЕЛЬНОСТИ (ПО ПОЛИНГУ)**

Сr	K	Na	Ca	Mg	Al	B	P	S	C	S	I	Br	Cl	N	O	F
6,8	8,8	6,9	1,0	1,2	1,8	2,0	2,1	2,1	2,5	2,5	2,8	2,8	3,0	3,0	3,5	4,0

**РАСТВОРЕНИЕ КИСЛОТ, СОЛЕЙ И ОСАЖЕНИЯ В ВОДЕ**

Вещь	H	Br	K	Na	Ag	Ba	Ca	Mg	Mn	Fe	Sn	Pb	Cu	Hg	Fe	Fe	Ni	Cd
ОН	Р	Р	Р	Р	—	Р	М	М	М	Н	Н	Н	Н	—	Н	Н	Н	Н
NO <sub>3</sub>	Р	Р	Р	Р	—	Р	М	М	М	Н	Н	Н	Н	—	Н	Н	Н	Н
ClO <sub>4</sub>	Р	Р	Р	Р	—	Р	М	М	М	Н	Н	Н	Н	—	Н	Н	Н	Н
Cl	Р	Р	Р	Р	—	Р	М	М	М	Н	Н	Н	Н	—	Н	Н	Н	Н
Br	Р	Р	Р	Р	—	Р	М	М	М	Н	Н	Н	Н	—	Н	Н	Н	Н
I	Р	Р	Р	Р	—	Р	М	М	М	Н	Н	Н	Н	—	Н	Н	Н	Н
S	Р	Р	Р	Р	—	—	—	—	—	Н	Н	Н	Н	—	Н	Н	Н	Н
SO <sub>4</sub>	Р	Р	Р	Р	—	М	М	М	М	Н	Н	Н	Н	—	Н	Н	Н	Н
CO <sub>3</sub>	Р	Р	Р	Р	—	М	М	М	М	Н	Н	Н	Н	—	Н	Н	Н	Н
PO <sub>4</sub>	Р	Р	Р	Р	—	М	М	М	М	Н	Н	Н	Н	—	Н	Н	Н	Н
NO <sub>2</sub>	Р	Р	Р	Р	—	М	М	М	М	Н	Н	Н	Н	—	Н	Н	Н	Н
CH <sub>3</sub> COO	Р	Р	Р	Р	—	М	М	М	М	Н	Н	Н	Н	—	Н	Н	Н	Н

**РЯД АКТИВНОСТИ МЕТАЛЛОВ / ЭЛЕКТРОНИЧЕСКИЙ РЯД НАПРЯЖЕНИЙ**

Li   K   Rb   Cs   Ba   Sr   Ca   Mg   Be   Al   Mn   Zn   Cd   Fe   Co   Ni   Sn   Pb   H<sub>2</sub>(HCl)   Cu   Hg   Ag   Pt   Au



Рис. 13. Таблица



Рис. 14. Схема

Сигналами (жест, световой сигнал, звуковой сигнал);

### ***Прием и передача информации в природе***

Живая природа сложна и разнообразна. Источниками и приемниками информации в ней являются живые организмы и их клетки. Организм обладает рядом свойств, отличающих его от неживых материальных объектов. Основные из них:

Непрерывный обмен веществом, энергией и информацией с окружающей средой;

Раздражимость (способность организма воспринимать и перерабатывать информацию об изменениях окружающей среды и внутренней среды организма);

Возбудимость (способность реагировать на действия раздражителей);

Самоорганизация (изменение организма для адаптации к условиям внешней среды).

До появления вычислительных машин наука биология, занимающаяся исследованиями живых организмов, давала описательные модели.

### ***Как происходит получение, обработка и накопление информации растениями?***

Растения должны выставить сигналы, которые делали бы их цветки особенно заметными. Важно также каким-то образом отметить цветки, которые уже опылены и не содержат более нектара. Это избавит насекомое от напрасной работы.

От вида опылителя зависит выбор растениями соответствующего сигнала.

Цветок, опыляемый только птицами, не должен привлекать насекомых.

### ***Как передают информацию пчелы?***



Рис. 15. Пчела

Удивителен способ передачи информации у пчел - это язык танца. Пчела, нашедшая цветущую поляну, прилетает к улью и начинает танцевать в воздухе перед собратьями, после чего пчелиный рой отправляется в указанное место за нектаром.

## **6. Хранение информации**

Смысл действия - сбросить зафиксированную (представленную) на носителе информацию (данные).

Память человека делится на:

Внутреннюю (оперативную) память. - Каждый человек хранит определенную информацию в собственной памяти - "в уме".

Внешнюю (долговременную) память (книги, флеш-память и т.п.). - Люди понимали ненадежность человеческой памяти и стремились зафиксировать наиболее важную информацию на внешних носителях.

## **7. Передача информации**

Смысл действия - поделиться, обменяться информацией, например, с другими людьми.

Любой процесс передачи информации упрощенно можно представить в виде:



В передаче информации всегда участвуют две стороны: тот, кто передает информацию (источник информации), и тот, кто получает информацию (приемник информации).

Передача информации осуществляется устно, письменно, по телефонным проводам, с помощью компьютерных сетей.

## **8. Обработка информации**

Обработка информации - это решение некоторой информационной задачи.

Смысл действия - изменить форму и смысл сообщения, получить новую информацию.

Пример: решение математической задачи.

Исполнителем, обрабатывающим информацию, является человек или специальное техническое устройство (например, компьютер).

## **9. Преобразование информации**

Смысл действия - изменить (подобрать) форму представления для хранения, использования, передачи, обработки информации.

Пример (Рис. 16) преобразование графической информации (рисунка) в текстовую информацию (текст):

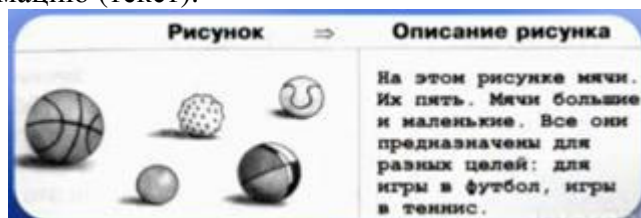


Рис. 16. Преобразование информации

### **3.1.3 Результаты и выводы:**

В ходе выполнения практической работы были изучены следующие вопросы

1. Что такое информация?

2. Какие действия человек совершает с информацией?
3. Как человек хранит информацию?
4. Носители информации.
5. Формы представления информации.

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.

### **3.2 Практическое занятие №2 ( 2 часа).**

**Тема:** «Элементы теории информации»

#### **3.2.1 Задание для работы:**

1. Решить задачу. В классе 30 человек. За контрольную работу по информатике получено 15 пятерок, 6 четверок, 8 троек и 1 двойка. Какое количество информации несет сообщение о том, что Андреев получил пятерку?
2. Решить задачу. В коробке лежат 64 цветных карандаша. Сообщение о том, что достали белый карандаш, несет 4 бита информации. Сколько белых карандашей было в корзине
3. Решить задачу. Информационное сообщение объемом 4 Кбайта содержит 4096 символов. Сколько символов содержит алфавит, при помощи которого было записано это сообщение?
4. Решить задачу. Пользователь вводит текст с клавиатуры со скоростью 90 знаков в минуту. Какое количество информации будет содержать текст, который он набирал 15 минут (используется компьютерный алфавит)

#### **3.2.2 Краткое описание проводимого занятия:**

##### **1.Вероятностный подход к определению количества информации.**

Для вычисления количества информации в сообщении о неравновероятном событии используют следующую формулу:  $I = \log_2(1/p)$

где  $I$  – это количество информации,  $p$  – вероятность события.

Вероятность события выражается в долях единицы и вычисляется по формуле:  $p = K/N$ ,

где  $K$  – величина, показывающая сколько раз произошло интересующее нас событие,  $N$  – общее число возможных исходов какого-то процесса.

рассмотрим задачу.

Пусть  $K_1$  – это количество пирожков с повидлом,  $K_1 = 24$

$K_2$  – количество пирожков с капустой,  $K_2 = 8$

$N$  – общее количество пирожков,  $N = K_1 + K_2 = 24 + 8 = 32$

Вычислим вероятность выбора пирожка с разной начинкой и количество информации, которое при этом было получено.

Вероятность выбора пирожка с повидлом:  $p_1 = 24/32 = 3/4 = 0,75$ .

Вероятность выбора пирожка с капустой:  $p_2 = 8/32 = 1/4 = 0,25$ .

Обращаем внимание учащихся на то, что в сумме все вероятности дают 1.

Вычислим количество информации, содержащееся в сообщении, что Маша выбрала пирожок с повидлом:  $I_1 = \log_2(1/p_1) = \log_2(1/0,75) = \log_2 1,3 = 1,15470$  бит.

Вычислим количество информации, содержащееся в сообщении, если был выбран пирожок с капустой:  $I_2 = \log_2(1/p_2) = \log_2(1/0,25) = \log_2 4 = 2$  бит.

При сравнении результатов вычислений получается следующая ситуация: вероятность выбора пирожка с повидлом больше, чем с капустой, а информации при этом получилось меньше. Это не случайность, а закономерность.

Качественную связь между вероятностью события и количеством информации в сообщении об этом событии можно выразить так: **чем меньше вероятность некоторого события, тем больше информации содержит сообщение об этом событии.**

Вернемся к нашей задаче с пирожками. Мы еще не ответили на вопрос: сколько получим информации при выборе пирожка любого вида?

Ответить на этот вопрос нам поможет формула вычисления количества информации для событий с различными вероятностями, которую предложил в 1948 г. американский инженер и математик К.Шеннон.

Если  $I$ -количество информации,  $N$ -количество возможных событий,  $p_i$  - вероятности отдельных событий, где  $i$  принимает значения от 1 до  $N$ , то количество информации для событий с различными вероятностями можно определить по формуле:

$$I = - \sum_{i=1}^N p_i \cdot \log_2 p_i$$

можно расписать формулу в таком виде:

$$I = -(p_1 \cdot \log_2 p_1 + p_2 \cdot \log_2 p_2 + p_3 \cdot \log_2 p_3 + p_4 \cdot \log_2 p_4 + \dots + p_N \cdot \log_2 p_N)$$

Рассмотрим формулу на нашем примере:

$$I = - (p_1 \cdot \log_2 p_1 + p_2 \cdot \log_2 p_2) = - (0,25 \cdot \log_2 0,25 + 0,75 \cdot \log_2 0,75) \approx - (0,25 \cdot (-2) + 0,75 \cdot (-0,42)) = 0,815 \text{ бит}$$

Можно ли применить формулу К. Шеннона для равновероятных событий?

Если  $p_1 = p_2 = \dots = p_n = 1/N$ , тогда формула принимает вид:

$$I = - \sum_{i=1}^N 1/N \cdot \log_2 1/N = \log_2 N$$

Мы видим, что формула Хартли является частным случаем формулы Шеннона.

### **Закрепление изучаемого материала.**

**Задача:** В корзине лежат 32 клубка красной и черной шерсти. Среди них 4 клубка красной шерсти.

Сколько информации несет сообщение, что достали клубок красной шерсти?  
Сколько информации несет сообщение, что достали клубок шерсти любой окраски?

**Дано:**  $K_k=4; N=32$

**Найти:**  $I_k, I$

**Решение:**

1. Найдем количество клубков черной шерсти:  $K_q = N - K_k$ ;  $K_q = 32 - 4 = 28$

2. Найдем вероятность доставания клубка каждого вида:  $p_k = K_k/N = 4/32 = 1/8$ ;  $p_q = K_q/N = 28/32 = 7/8$ ;

3. Найдем количество информации, которое несет сообщение, что достали клубок красной шерсти:  $I_k = \log_2(1/(1/p_k)) = \log_2(1/(1/8)) = \log_2 8 = 3 \text{ бит}$

4. Найдем количество информации, которое несет сообщение, что достали клубок шерсти любой окраски:

$$I = - \sum_{i=1}^N p_i \cdot \log_2 p_i = - (1/8 \cdot \log_2 (1/8) + 7/8 \cdot \log_2 (7/8)) = - ((1/8) \cdot (-3) + (7/8) \cdot (-0,193)) = - (-0,375 - 0,169) = 0,547 \text{ бит}$$

Ответ:  $I_k = 3 \text{ бит}$ ;  $I = 0,547 \text{ бит}$

### **2.Алфавитный подход к определению количества информации**



При хранении и передаче информации с помощью технических устройств информацию следует рассматривать как последовательность символов - знаков (букв, цифр, кодов цветов точек изображения и т.д.).

Набор символов знаковой системы (алфавит) можно рассматривать как различные возможные состояния (события).

Тогда, если считать, что появление символов в сообщении равновероятно, количество возможных событий  $N$  можно вычислить как  $N=2^i$ . Количество информации в сообщении  $I$  можно подсчитать умножив количество символов  $K$  на информационный вес одного символа  $i$ .

Итак, мы имеем формулы, необходимые для определения количества информации в алфавитном подходе:

$N=2^i$	$i$	Информационный вес символа, бит
	$N$	Мощность алфавита
$=K*i$	$K$	Количество символов в тексте
	$I$	Информационный объем текста

### Закрепление изучаемого материала.

Задача. Сообщение, составленное с помощью 32 – символьного алфавита, содержит 80 символов. Другое сообщение составлено с использованием 64 – символьного алфавита и содержит 70 символов. Сравните объемы информации, содержащейся в сообщениях.

**Дано:** :  $N_1 = 32$ ,  $K_1 = 80$ ,  $N_2 = 64$ ,  $K_2 = 70$

**Найти:**  $I_{t1}$ ,  $I_{t2}$

**Решение:**

- 1)  $I_t = K*i$ , где  $i$  – объем одного символа
  - 2)  $2^i = N$ ,  $2^i = 32$ ,  $i = 5$  бит – объем одного символа первого сообщения;
  - 3)  $2^i = N$ ,  $2^i = 64$ ,  $i = 6$  бит – объем одного символа второго сообщения;
  - 4)  $I_{t1} = K_1 * i = 80 * 5 = 400$  бит – объем первого сообщения;
  - 5)  $I_{t2} = K_2 * i_2 = 70 * 6 = 420$  бит – объем второго сообщения;
- Ответ: во втором сообщении информации больше, чем в первом.

### 3.2.3 Результаты и выводы:

В ходе выполнения практической работы студентам были предложены перечень задач по теме практической работы.

На занятии применялись разнообразные формы работы: фронтальная, групповая, индивидуальная. Фронтальная работа проводилась на этапе актуализации по обобщению и систематизации знаний – ребята отвечали на поставленные мной вопросы. Групповая форма работы использовалась на этапе закрепления темы и решения задач у доски. Индивидуальная форма работы - выполнение практической работы. При затруднении выполнения работы мною оказывалась помощь учащимся.

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.

### 3.3 Практическое занятие №3 ( 2 часа).

**Тема:** «Позиционные и непозиционные системы счисления»

### 3.3.1 Задание для работы:

1. Теоретическая часть
2. Перевод целых чисел из одной системы счисления в другую
3. Решение задач

### 3.3.2 Краткое описание проводимого занятия:

#### 1. Теоретическая часть

##### Позиционные и непозиционные системы счисления

Системой счисления называется совокупность приемов наименования и записи чисел. В любой системе счисления числа записываются как последовательность цифр. Такие системы подразделяются на позиционные и непозиционные.

Если смысловое значение цифры не зависит от ее места в последовательности, то такая система счисления называется **непозиционной**. Примером непозиционной системы счисления является римская система счисления. В числах IX и XI присутствует цифра I. В первом случае она стоит на нулевой позиции, а во втором случае на первой позиции. Но в какой бы позиции она не стояла, ее смысловое значение равно единице.

Наоборот, если смысловое значение цифры зависит от ее места в последовательности, то такая система счисления называется **позиционной**. Примером позиционной системы счисления является всем нам хорошо известная десятичная система счисления. Любое число в ней представляется с помощью набора из десяти цифр: 0, 1, 2, ..., 9. Эти числа называются базисными. Например, число 777 состоит из трех семерок. Но в каждой позиции каждая цифра имеет различный смысл. Самая правая цифра 7 говорит о количестве единиц в числе, следующая – о количестве десятков, еще следующая – о количестве сотен. Все это можно выразить следующим образом:

$$777 = 7 * 10^2 + 7 * 10^1 + 7 * 10^0.$$

Само число 10 называется основанием десятичной системы счисления.

В современной вычислительной технике, в устройствах автоматики и связи широко используется двоичная система счисления. В ней для изображения числа используются только две цифры: 0 и 1, которые являются базисными цифрами, а цифра 2 – основание двоичной системы счисления. Также при построении ЭВМ можно использовать элементы, которые могут находиться только в двух состояниях. Например, высокое или низкое напряжение в цепи, наличие или отсутствие электрического импульса и т. п. Это обстоятельство, а также простота выполнения арифметических операций являются причиной того, что большинство современных ЭВМ являются причиной того, что большинство современных ЭВМ используют двоичную систему счисления.

Существует восьмеричная система счисления, в которой базисными цифрами являются 0, 1, 2, ..., 7, а основанием цифра 8.

В шестнадцатеричной системе счисления основанием является цифра 16, а базисными цифрами: цифры 0, 1, 2, ..., 9 и буквы латинского алфавита: цифре 10 соответствует буква A, 11 – B, 12 – C, 13 – D, 14 – E, 15 – F.

#### 2. Перевод целых чисел из одной системы счисления в другую

Возможен перевод чисел из десятичной системы счисления в двоичную. Для этого надо число, заданное в десятичной системе счисления, разделить на основание двоичной системы счисления 2. Причем деление производить до тех пор, пока частное не станет меньше делителя, а получившиеся остатки записать в обратном порядке.

При обратном переводе используется метод, базирующийся на умножении цифр переводимого числа на основание двоичной системы счисления в степени  $q$ , где  $q$  – порядковый номер разряда.

### 3. Решение задач

Задача Перевести число 614 из десятичной системы счисления в двоичную.

Для этого число  $614_{10}$  делим на основание двоичной системы счисления – цифру 2. Полученное частное делим до тех пор, пока оно не станет меньше делителя (основания двоичной системы счисления – 2). Первой цифрой искомого числа является последнее частное, а остальные цифры – это остатки, полученные от деления, т. е.  $614_{10} \rightarrow 1001100110_2$ .

Задача Перевести двоичное число  $1001101_2$  в десятичную систему счисления.

Применим следующий метод:

$$1^6 0^5 0^4 1^3 1^2 0^1 1^0_2 = 1 * 2^6 + 0 * 2^5 + 0 * 2^4 + 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = \\ = 1 + 4 + 8 + 64 = 77_{10}.$$

Перевод из десятичной системы счисления в восьмеричную аналогичен переводу из десятичной системы счисления в двоичную. А перевод из восьмеричной системы счисления в десятичную производится по тому же правилу, что и перевод из двоичной системы счисления в десятичную. Так при переводе числа  $614_{10}$  в восьмеричную систему счисления получаем число  $1146_8$ .

Перевод из десятичной системы счисления в шестнадцатеричную и наоборот выполняется по тем же правилам, описанным выше. При переводе числа  $614_{10}$  получаем число  $266_{16}$ .

#### 3.3.3 Результаты и выводы:

В ходе выполнения практической работы студентам были предложены перечень задач по теме практической работы.

На занятии применялись разнообразные формы работы: фронтальная, групповая, индивидуальная. Фронтальная работа проводилась на этапе актуализации по обобщению и систематизации знаний – ребята отвечали на поставленные мной вопросы. Групповая форма работы использовалась на этапе закрепления темы и решения задач у доски. Индивидуальная форма работы - выполнение практической работы. При затруднении выполнения работы мною оказывалась помощь учащимся.

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.

### 3.4 Практическое занятие №4 ( 2 часа).

**Тема:** «Перевод чисел из одной системы счисления в другую систему счисления»

#### 3.4.1 Задание для работы:

1. Перевод чисел в десятичную систему
2. Перевод целых десятичных чисел в недесятичную систему
3. Перевод правильных дробей из десятичной системы счисления в недесятичную

#### 3.4.2 Краткое описание проводимого занятия:

**1. Перевод чисел в десятичную систему** осуществляется путем составления степенного ряда с основанием той системы, из которой число переводится. Затем подсчитывается значение суммы.

Пример.

а) Перевести  $10101101.101_2$  "10" с.с.

*Здесь и в дальнейшем при одновременном использовании нескольких различных систем счисления основание системы к которой относится число будем указывать в виде нижнего индекса.*

$$10101101.101_2 = 12^7 + 02^6 + 12^5 + 02^4 + 12^3 + 12^2 + 02^1 + 12^0 + 12^{-1} + 02^{-2} + 12^{-3} = 173.625_{10}$$

б) Перевести  $703.04_8$  "10" с.с.

$$703.04_8 = 78^2 + 08^1 + 38^0 + 08^{-1} + 48^{-2} = 451.0625_{10}$$

в) Перевести  $B2E.4_{16}$  "10" с.с.

$$B2E.4_{16} = 1116^2 + 216^1 + 1416^0 + 416^{-1} = 2862.25_{10}$$

**2.Перевод целых десятичных чисел в недесятичную систему счисления** осуществляется последовательным делением десятичного числа на основание той системы, в которую оно переводится, до тех пор, пока не получится частное меньшее этого основания. Число в новой системе записывается в виде остатков деления, начиная с последнего.

Пример.

а) Перевести  $181_{10}$  "8" с.с.

$$\begin{array}{r|l} 181 & 8 \\ -176 & 22 \\ \hline 5 & 16 \\ -5 & 16 \\ \hline 0 & 2 \end{array}$$

Результат:  $181_{10} = 265_8$

б) Перевести  $622_{10}$  "16" с.с.

$$\begin{array}{r|l} 622 & 16 \\ -48 & 38 \\ \hline 142 & 32 \\ -128 & 14 \\ \hline 14 & 6 \end{array}$$

Результат:  $622_{10} = 26E_{16}$

**3.Перевод правильных дробей из десятичной системы счисления в недесятичную.**

Для перевода правильной десятичной дроби в другую систему эту дробь надо последовательно умножать на основание той системы, в которую она переводится. При этом умножаются только дробные части. Дробь в новой системе записывается в виде целых частей произведений, начиная с первого.

Пример.

Перевести  $0.3125_{10}$  "8" с.с.

$$\begin{array}{l|l} 0 & 3125 \times 8 \\ \hline 2 & 5000 \times 8 \\ \hline 4 & 0000 \end{array}$$

Результат:  $0.3125_{10} = 0.24_8$

**Замечание.** Конечной десятичной дроби в другой системе счисления может соответствовать бесконечная (иногда периодическая) дробь. В этом случае количество знаков в представлении дроби в новой системе берется в зависимости от требуемой точности.

Пример.

Перевести  $0.65_{10}$  "2" с.с. Точность 6 знаков.

	0	65 × 2
	1	3 × 2
	0	6 × 2
	1	2 × 2
	0	4 × 2
	0	8 × 2
	1	6 × 2
	...	

Результат:  $0.65_{10} \ 0.10(1001)_2$

Для перевода неправильной десятичной дроби в систему счисления с недесятичным основанием необходимо отдельно перевести целую часть и отдельно дробную.

Пример.

Перевести  $23.125_{10}$  "2" с.с.

1) Переведем целую часть:      2) Переведем дробную часть:

23	2
22	11 2
1	10 5 2
	1 4 2 2
	1 2 1
	0

0	125 × 2
0	25 × 2
0	5 × 2
1	0

Таким образом:  $23_{10} = 10111_2$ ;  $0.125_{10} = 0.001_2$ .

Результат:  $23.125_{10} = 10111.001_2$ .

Необходимо отметить, что целые числа остаются целыми, а правильные дроби - дробями в любой системе счисления.

Для перевода восьмеричного или шестнадцатеричного числа в двоичную форму достаточно заменить каждую цифру этого числа соответствующим трехразрядным двоичным числом (триадой) (Таб. 1) или четырехразрядным двоичным числом (тетрадой) (Таб. 1), при этом отбрасывают ненужные нули в старших и младших разрядах.

Пример.

а) Перевести  $305.4_8$  "2" с.с.

$$\begin{array}{cccc} \underline{3} & \underline{0} & \underline{5} & \underline{.4} \\ \underline{011} & \underline{000} & \underline{101} & \underline{100} \end{array} = 11000101.1_2$$

б) Перевести  $7B2.E_{16}$  "2" с.с.

$$\begin{array}{cccc} \underline{7} & \underline{B} & \underline{2} & \underline{.E} \\ \underline{0111} & \underline{1011} & \underline{0010} & \underline{1110} \end{array} = 11110110010.111_2$$

Для перехода от двоичной к восьмеричной (шестнадцатеричной) системе поступают следующим образом: двигаясь от точки влево и вправо, разбивают двоичное число на группы по три (четыре) разряда, дополняя при необходимости нулями крайние левую и правую группы. Затем триаду (тетраду) заменяют соответствующей восьмеричной (шестнадцатеричной) цифрой.

Пример.

а) Перевести  $1101111001.1101_2$  "8" с.с.

$$\begin{array}{cccccc} \underline{001} & \underline{101} & \underline{111} & \underline{001} & \underline{.110} & \underline{100} \\ \underline{1} & \underline{5} & \underline{7} & \underline{1} & \underline{6} & \underline{4} \end{array} = 1571.64_8$$

б) Перевести  $1111111011.100111_2$  "16" с.с.

$$\underbrace{01111111}_{7} \underbrace{1011}_F \underbrace{.100111}_{9} \underbrace{100}_C = 7FB.9C_{16}$$

Перевод из восьмеричной в шестнадцатеричную систему и обратно осуществляется через двоичную систему с помощью триад и тетрад.

Пример. Перевести  $175.24_8$  "16" с.с.

$$\underbrace{1}_{001} \underbrace{7}_{111} \underbrace{5}_{101} \underbrace{.2}_{010} \underbrace{4}_{100}_8 = 1111101.0101_2 = \underbrace{01111101}_7 \underbrace{.0101}_D_2 = 7D.5_{16}$$

Результат:  $175.24_8 = 7D.5_{16}$ .

### 3.4.3 Результаты и выводы:

В ходе выполнения практической работы студентам были предложены перечень задач по теме практической работы.

На занятии применялись разнообразные формы работы: фронтальная, групповая, индивидуальная. Фронтальная работа проводилась на этапе актуализации по обобщению и систематизации знаний – ребята отвечали на поставленные мной вопросы. Групповая форма работы использовалась на этапе закрепления темы и решения задач у доски. Индивидуальная форма работы - выполнение практической работы. При затруднении выполнения работы мною оказывалась помощь учащимся.

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.

## 3.5 Практическое занятие №5 ( 4 часа).

**Тема:** «Понятие и свойства алгоритма. Базовые алгоритмические структуры»

### 3.5.1 Задание для работы:

1. Развитие знаний по составлению алгоритмов с использованием различных структур
2. Развитие познавательного интереса, логического мышления, речи и внимания учащихся, формирование информационной культуры и потребности приобретения знаний;

### 3.5.2 Краткое описание проводимого занятия:

#### Теоретический материал:

**АЛГОРИТМ** - это последовательность команд, ведущих к какой-либо цели.

Это строго определенная процедура, гарантирующая получение результата за конечное число шагов. Это правило, указывающее действия, в результате цепочки которых происходит переход от исходных данных к искомому результату. Указанная цепочка действий называется алгоритмическим процессом, а каждое отдельное действие - его шагом. Пример: площадь прямоугольника  $S=a \cdot b$ .

Виды алгоритмов: вычислительные, диалоговые, графические, обработки данных, управления объектами и процессами и др.

Свойства алгоритмов - однозначность (и определенность), результативность (и выполнимость), правильность (и понятность), массовость или универсальность (т.е. применимость для целого класса задач, к различным наборам исходных данных).

Способы записи алгоритмов:

В виде блок-схем, в виде программ, в виде текстовых описаний (рецепты, например, рецепты приготовления пищи, лекарств и др.).

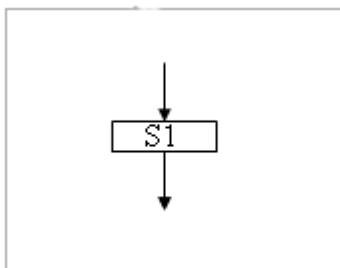
По вариантам написать предложенные преподавателем алгоритмы при помощи различных способах записи алгоритмов и при помощи различных структур.

**Контрольные вопросы:**

1. Что такое алгоритм?
2. Какие способы записи алгоритмов вы знаете?
3. Какие свойства алгоритмов Вам известны?
4. Составьте алгоритм приготовления любого блюда?
5. Постройте блок-схему на составленный алгоритм?

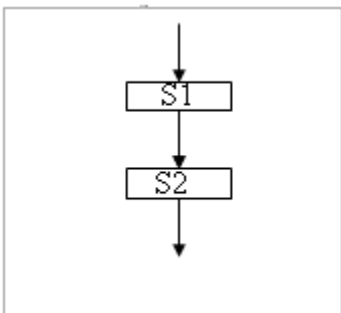
Наиболее понятно структуру алгоритма можно представить с помощью блок-схемы, в которой используются геометрические фигуры (блоки), соединенные между собой стрелками, указывающими последовательность выполнения действий. Приняты определенные стандарты графических изображений блоков. Например, команду обработки информации помещают в блок, имеющий вид прямоугольника, проверку условий - в ромб, команды ввода или вывода - в параллелограмм, а овалом обозначают начало и конец алгоритма.

Структурной элементарной единицей алгоритма является простая команда, обозначающая один элементарный шаг переработки или отображения информации. Простая команда на языке схем изображается в виде функционального блока.



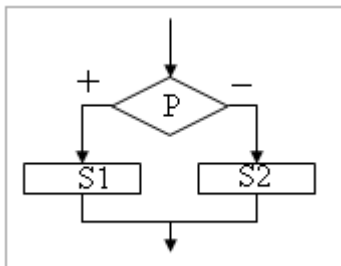
Данный блок имеет *один вход* и *один выход*. Из простых команд и проверки условий образуются составные команды, имеющие более сложную структуру и тоже *один вход* и *один выход*.

Структурный подход к разработке алгоритмов определяет использование только базовых алгоритмических структур (конструкций): следование, ветвление, повторение, которые должны быть оформлены стандартным образом.

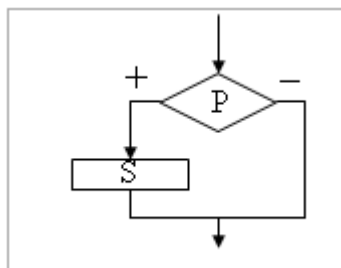


Рассмотрим основные структуры алгоритма.

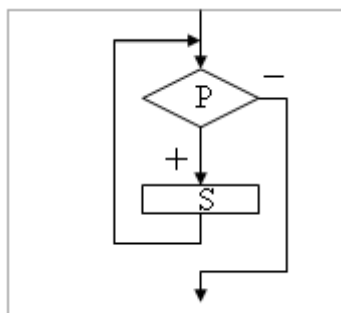
Команда *следования* состоит только из простых команд. На рисунке простые команды имеют условное обозначение *S1* и *S2*. Из команд следования образуются линейные алгоритмы. Примером линейного алгоритма будет нахождение суммы двух чисел, введенных с клавиатуры.



Команда *ветвления* - это составная команда алгоритма, в которой в зависимости от условия *P* выполняется или одно *S1*, или другое *S2* действие. Из команд следования и команд ветвления составляются разветвляющиеся алгоритмы (алгоритмы ветвления). Примером разветвляющегося алгоритма будет нахождение большего из двух чисел, введенных с клавиатуры.



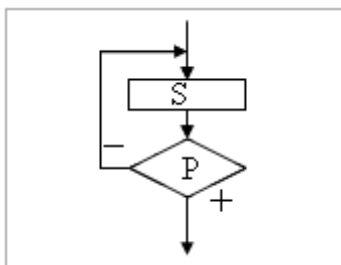
Команда ветвления может быть полной и неполной формы. Неполная форма команды ветвления используется тогда, когда необходимо выполнять действие *S* только в случае соблюдения условия *P*. Если условие *P* не соблюдается, то команда ветвления завершает свою работу без выполнения действия. Примером команды ветвления неполной формы будет уменьшение в два раза только четного числа.



Команда *повторения* - это составная команда алгоритма, в которой в зависимости от условия *P* возможно многократное выполнение действия *S*. Из команд следования и команд повторения составляются циклические алгоритмы (алгоритмы повторения). На рисунке представлена команда повторения с предусловием. Называется она так потому, что вначале проверяется условие, а уже затем выполняется действие. Причем действие выполняется, пока условие соблюдается. Пример циклического алгоритма может быть следующий. Пока с клавиатуры вводятся положительные числа, алгоритм выполняет нахождение их суммы.



Команда повторения с предусловием не является единственно возможной. Разновидностью команды повторения с предусловием является команда повторения с параметром. Она используется тогда, когда известно количество повторений действия. В блок-схеме команды повторения с параметром условие записывается не в ромбе, а в шестиугольнике. Примером циклического алгоритма с параметром будет нахождение суммы первых 20 натуральных чисел.



В команде повторения с постусловием вначале выполняется действие  $S$  и лишь затем, проверяется условие  $P$ . Причем действие повторяется до тех пор, пока условие не соблюдается. Примером команды повторения с постусловием будет уменьшение положительного числа до тех пор, пока оно неотрицательное. Как только число становится отрицательным, команда повторения заканчивает свою работу.

С помощью соединения только этих элементарных конструкций (последовательно или вложением) можно "собрать" алгоритм любой степени сложности.

### Линейный алгоритм

Приведем пример записи алгоритма в виде блок-схемы, псевдокодов и на языке Паскаль. Ручное тестирование и подбор системы тестов выполняются аналогично предыдущему заданию.

Блок-схема	Псевдокоды	Паскаль
	<p><b>алг</b> среднее геометрическое</p> <p><b>вещ</b> a, b, g</p> <p><b>нач</b></p> <p><b>ввод</b> a, b</p> <p><math>g := (a * b)^{(1/2)}</math></p> <p><b>вывод</b> g</p> <p><b>кон</b></p>	<pre> <b>program</b> Srednee_geometr;   <b>var</b> a, b, g: real; <b>begin</b>     <b>readln</b> (a, b);     s := sqrt(a * b);     <b>writeln</b> (g) <b>end.</b> </pre>

1. Построить линейный алгоритм вычисления значения  $Y$  по формуле  $Y=(7X+4)(2X-2)$  при  $X=3$ . Составьте алгоритм самостоятельно, выделяя каждое действие как отдельный шаг.
2. В какой форме записываются алгоритмы? Напишите вывод.

### 3.5.3 Результаты и выводы:

В ходе выполнения практической работы студентам были предложены перечень задач по теме практической работы.

На занятии применялись разнообразные формы работы: фронтальная, групповая, индивидуальная. Фронтальная работа проводилась на этапе актуализации по обобщению и систематизации знаний – ребята отвечали на поставленные мной вопросы. Групповая форма работы использовалась на этапе закрепления темы и решения задач у доски.

Индивидуальная форма работы - выполнение практической работы. При затруднении выполнения работы мною оказывалась помощь учащимся.

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.

### 3.6 Практическое занятие №6 ( 6 часов).

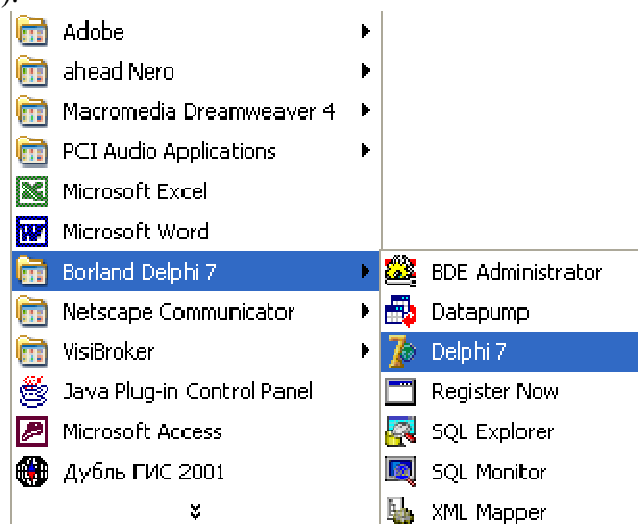
**Тема:** «Языки программирования высокого уровня»

#### 3.6.1 Задание для работы:

1. Развитие познавательного интереса, логического мышления.
2. Знакомство с операторами и функциями, и правилами написания программ в среде Delphi 7
3. Развитие алгоритмического мышления, памяти, внимательности.

#### 3.6.2 Краткое описание проводимого занятия:

Запускается Delphi обычным образом, т. е. выбором из меню **Borland Delphi 7** команды **Delphi 7** (рис. 1).



**Рис. 1** Запуск Delphi

Вид экрана после запуска Delphi несколько необычен (рис. 2). Вместо одного окна на экране появляются пять:

1. главное окно — **Delphi 7**;
2. окно стартовой формы — **Form 1**;
3. окно редактора свойств объектов — **Object Inspector**;
4. окно просмотра списка объектов — **Object TreeView**;
5. окно редактора кода — **Unit1.pas**.

Окно редактора кода почти полностью закрыто окном стартовой формы.

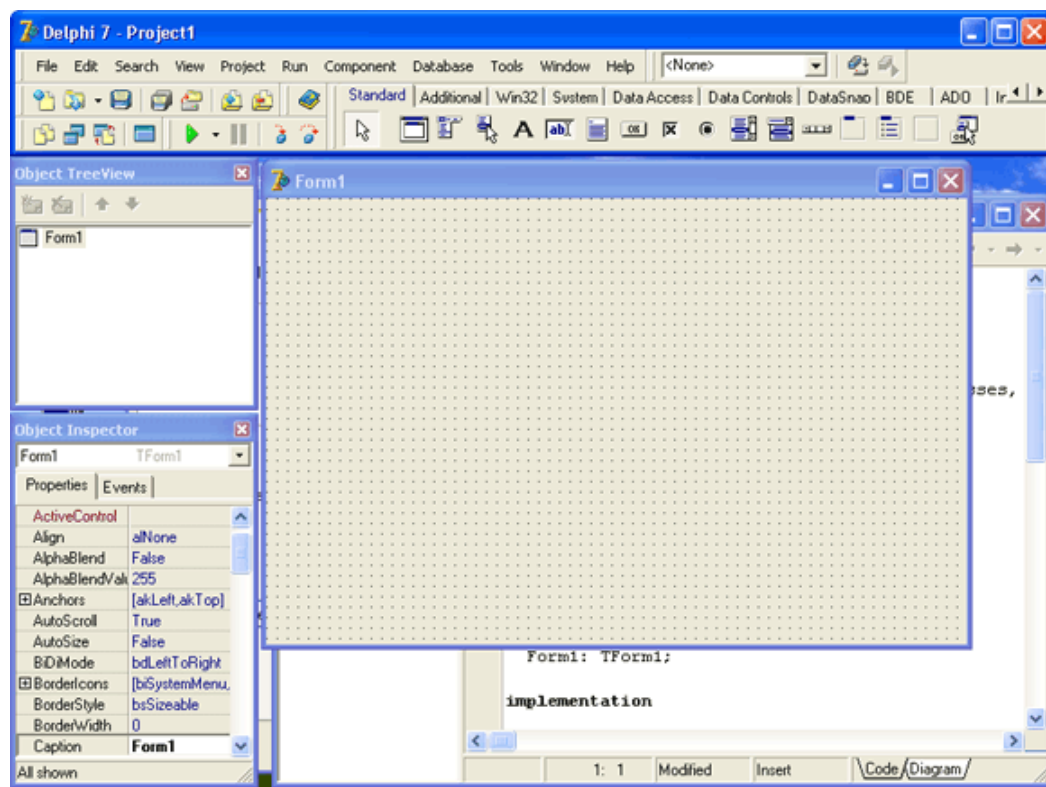


Рис. 2 Вид экрана после запуска Delphi

В главном окне (рис. 3) находится меню команд, панели инструментов и палитра компонентов.

Окно стартовой формы (**Form1**) представляет собой заготовку главного окна разрабатываемого приложения.

Программное обеспечение принято делить на системное и прикладное. Системное программное обеспечение — это все то, что составляет операционную систему. Остальные программы принято считать прикладными. Для краткости прикладные программы называют приложениями.

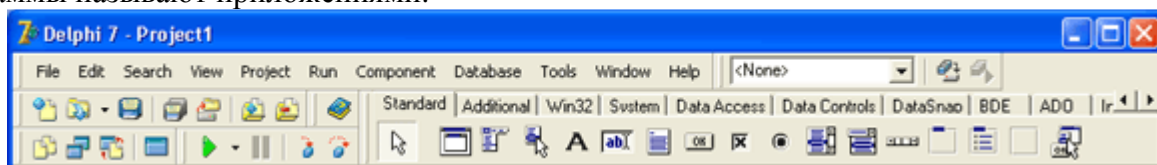
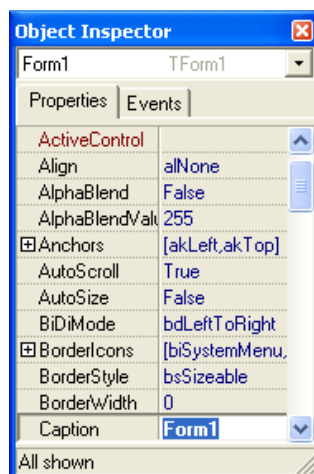
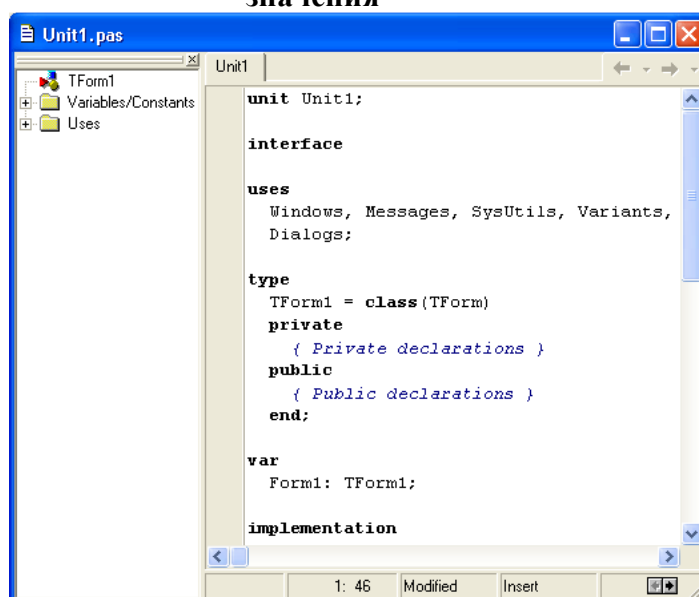


Рис. 3 Главное окно

Окно **Object Inspector** (рис. 4) — окно редактора свойств объектов предназначено для редактирования значений свойств объектов. В терминологии визуального проектирования *объекты* — это диалоговые окна и элементы управления (поля ввода и вывода, командные кнопки, переключатели и др.). *Свойства объекта* — это характеристики, определяющие вид, положение и поведение объекта. Например, свойства width и Height задают размер (ширину и высоту) формы, свойства top и Left — положение формы на экране, свойство caption — текст заголовка.



**Рис. 4** На вкладке **Properties** перечислены свойства объекта и указаны их значения

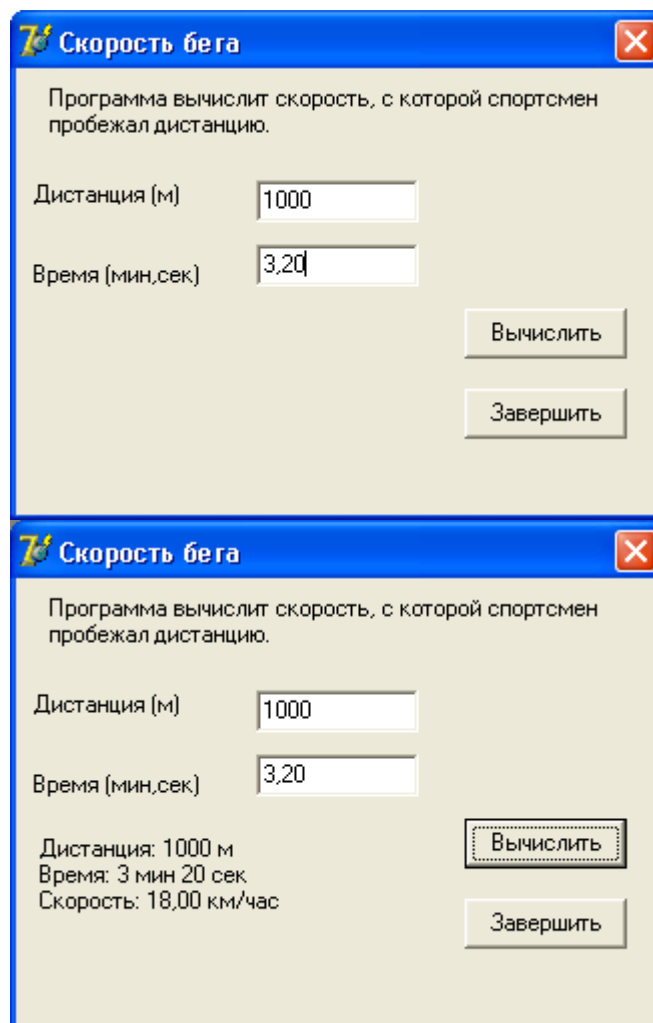


**Рис. 5** Окно редактора кода

В окне редактора кода (рис. 5), которое можно увидеть, отодвинув в сторону окно формы, следует набирать текст программы. В начале работы над новым проектом это окно редактора кода содержит сформированный Delphi шаблон программы.

### **Первый проект**

Для демонстрации возможностей Delphi и технологии визуального проектирования разработаем приложение, используя которое, можно вычислить скорость, с которой спортсмен пробежал дистанцию. Вид окна программы во время ее работы приведен на рис. 6.



**Рис. 6** Окно программы вычисления скорости бега

Для начала работы над новой программой запустите Delphi. Если вы уже работаете в среде разработки и у вас загружен другой проект, выберите в меню **File** (Файл) команду **New | Application** (Создать | Приложение).

### **Форма**

Работа над *новым проектом*, так в Delphi называется разрабатываемое приложение, начинается с создания стартовой формы. Так на этапе разработки программы называют диалоговые окна.

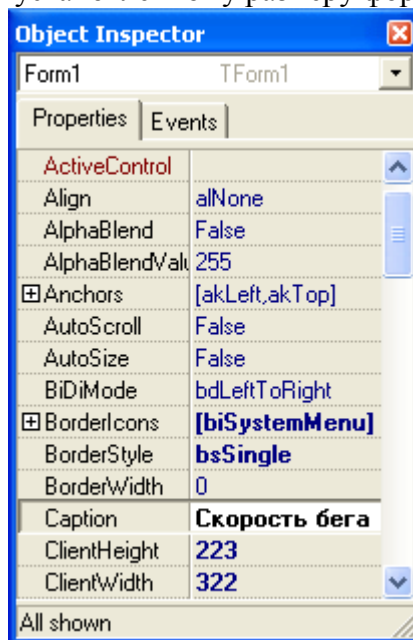
Стартовая форма создается путем изменения значений свойств формы **Form1** и добавления к форме необходимых компонентов (полей ввода и вывода текста, командных кнопок).

При создании формы в первую очередь следует изменить значение свойства **caption** (Заголовок). В нашем примере надо заменить текст **Form1** на "скорость бега". Чтобы это сделать, нужно в окне **Object Inspector** щелкнуть мышью на строке **Caption**, в результате чего будет выделено текущее значение свойства, в строке появится курсор, и можно будет ввести текст "скорость бега" (рис. В12).

Аналогичным образом можно установить значения свойств **Height** и **width**, которые определяют высоту и ширину формы. Размер формы и ее положение на экране, а также размер других элементов управления и их положение на поверхности формы задают в

пикселах, т. е. точках экрана. Свойствам Height и width надо присвоить значения 250 и 330 соответственно.

Форма — это обычное окно. Поэтому его размер можно изменить точно так же, как размер любого другого окна, т. е. захватом и перемещением (с помощью мыши) границы. По окончании перемещения границ автоматически изменятся значения свойств Height и width. Они будут соответствовать установленному размеру формы.

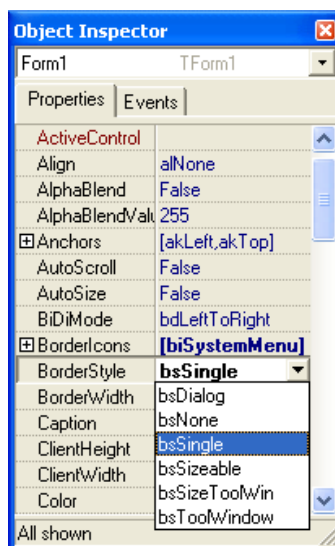


**Рис. В12.** Установка значения свойства путем ввода значения

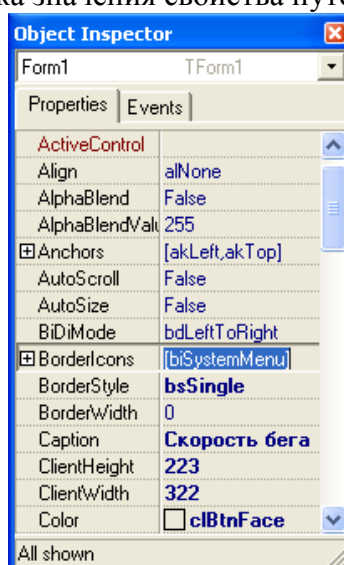
Положение диалогового окна на экране после запуска программы соответствует положению формы во время ее разработки, которое определяется значением свойств Top (отступ от верхней границы экрана) и Left (отступ от левой границы экрана). Значения этих свойств также можно задать путем перемещения окна формы при помощи мыши.

При выборе некоторых свойств, например, Borderstyle, справа от текущего значения свойства появляется значок раскрывающегося списка. Очевидно, что значение таких свойств можно задать путем выбора из списка (рис. В13).

Некоторые свойства являются сложными, т. е. их значение определяется совокупностью значений других (уточняющих) свойств. Перед именами сложных свойств стоит значок "+", при щелчке на котором раскрывается список уточняющих свойств (рис. В14). Например, свойство BorderIcons определяет, какие кнопки управления окном будут доступны во время работы программы. Так, если свойству biMaximize присвоить значение False, то во время работы программы кнопки **Развернуть** в заголовке окна не будет.



**Рис. В13.** Установка значения свойства путем выбора из списка



**Рис. В14.** Раскрытый список вложенных свойств сложного свойства BorderIcons

Рядом со значениями некоторых свойств отображается командная кнопка с тремя точками. Это значит, что для задания значения свойства можно воспользоваться дополнительным диалоговым окном. Например, значение сложного свойства Font можно задать путем непосредственного ввода значений уточняющих свойств, а можно воспользоваться стандартным диалоговым окном выбора шрифта.

### 3.6.3 Результаты и выводы:

В ходе выполнения практической работы студентам были предложены перечень задач по теме практической работы.

На занятии применялись разнообразные формы работы: фронтальная, групповая, индивидуальная. Фронтальная работа проводилась на этапе актуализации по обобщению и систематизации знаний – ребята отвечали на поставленные мной вопросы. Групповая форма работы использовалась на этапе закрепления темы и решения задач у доски. Индивидуальная форма работы - выполнение практической работы. При затруднении выполнения работы мною оказывалась помощь учащимся.

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.

### **3.7 Практическое занятие №7 ( 2 часа).**

**Тема:** «Операционные системы и среды»

#### **3.7.1 Задание для работы:**

1. Понятие архитектуры операционной системы
2. Монолитная архитектура
3. Структурированная архитектура
4. Многослойная структура ОС

#### **3.7.2 Краткое описание проводимого занятия:**

1. Понятие архитектуры операционной системы

Под архитектурой операционной системы понимают структурную и функциональную организацию ОС на основе некоторой совокупности программных модулей. В состав ОС входят исполняемые и объектные модули стандартных для данной ОС форматов, программные модули специального формата (например, загрузчик ОС, драйверы ввода-вывода), конфигурационные файлы, файлы документации, модули справочной системы и т.д.

На архитектуру ранних операционных систем обращалось мало внимания: во-первых, ни у кого не было опыта в разработке больших программных систем, а во-вторых, проблема взаимозависимости и взаимодействия модулей недооценивалась. В подобных монолитных ОС почти все процедуры могли вызывать одна другую. Такое отсутствие структуры было несовместимо с расширением операционных систем. Первая версия ОС OS/360 была создана коллективом из 5000 человек за 5 лет и содержала более 1 млн строк кода. Разработанная несколько позже операционная система Mastics содержала к 1975 году уже 20 млн строк. Стало ясно, что разработка таких систем должна вестись на основе модульного программирования.

Большинство современных ОС представляют собой хорошо структурированные модульные системы, способные к развитию, расширению и переносу на новые платформы. Какой-либо единой унифицированной архитектуры ОС не существует, но известны универсальные подходы к структурированию ОС. Принципиально важными универсальными подходами к разработке архитектуры ОС являются [5, 10, 13, 17]:

- модульная организация;
- функциональная избыточность;
- функциональная избирательность;
- параметрическая универсальность;
- концепция многоуровневой иерархической вычислительной системы, по которой ОС представляется многослойной структурой;
- разделение модулей на две группы по функциям: ядро – модули, выполняющие основные функции ОС, и модули, выполняющие вспомогательные функции ОС;
- разделение модулей ОС на две группы по размещению в памяти вычислительной системы: резидентные, постоянно находящиеся в оперативной памяти, и транзитные, загружаемые в оперативную память только на время выполнения своих функций;



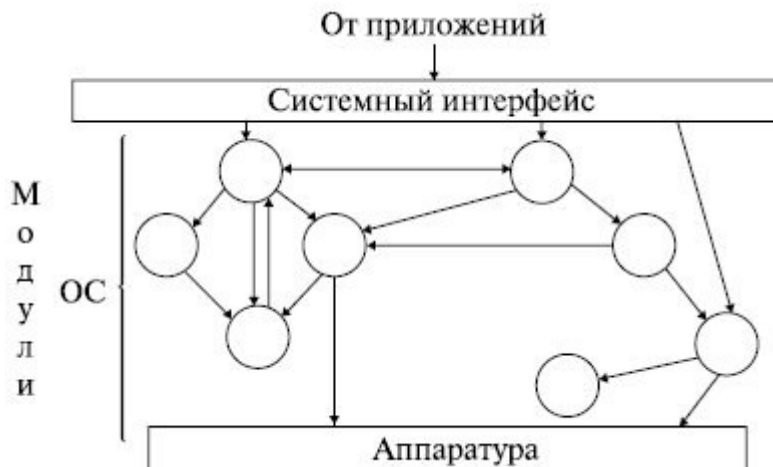
- реализация двух режимов работы вычислительной системы: привилегированного режима (режима ядра – Kernel mode), или режима супервизора (supervisor mode), и пользовательского режима (user mode), или режима задачи (task mode);
- ограничение функций ядра (а следовательно, и количества модулей ядра) до минимального количества необходимых самых важных функций.

## 2. Монолитная архитектура

Первые ОС разрабатывались как монолитные системы без четко выраженной структуры (рис. 1.2).

Для построения монолитной системы необходимо скомпилировать все отдельные процедуры, а затем связать их вместе в единый объектный файл с помощью компоновщика (примерами могут служить ранние версии ядра UNIX или Novell NetWare). Каждая процедура видит любую другую процедуру (в отличие от структуры, содержащей модули, в которой большая часть информации является локальной для модуля, и процедуры модуля можно вызвать только через специально определенные точки входа).

Однако даже такие монолитные системы могут быть немного структурированными. При обращении к системным вызовам, поддерживаемым ОС, параметры помещаются в строго определенные места, такие как регистры или стек, а затем выполняется специальная команда прерывания, известная как вызов ядра или вызов супервизора. Эта команда переключает машину из режима пользователя в режим ядра, называемый также режимом супервизора, и передает управление ОС. Затем ОС проверяет параметры вызова, для того чтобы определить, какой системный вызов должен быть выполнен. После этого ОС индексирует таблицу, содержащую ссылки на процедуры, и вызывает соответствующую процедуру.



**Рис. 1.2.** Монолитная архитектура

Такая организация ОС предполагает следующую структуру:

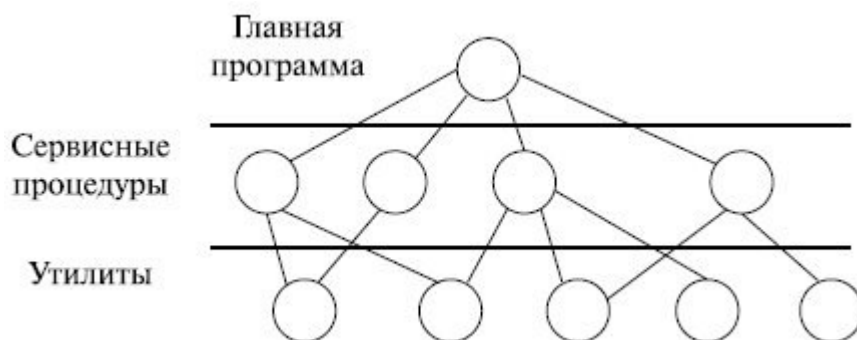
- главная программа, которая вызывает требуемые сервисные процедуры;
- набор сервисных процедур, реализующих системные вызовы;
- набор утилит, обслуживающих сервисные процедуры.

## 3. Структурированная архитектура

В этой модели для каждого системного вызова имеется одна сервисная процедура. Утилиты выполняют функции, которые нужны нескольким сервисным процедурам. Это деление процедур на три слоя показано на рис. 1.3.

Классической считается архитектура ОС, основанная на концепции иерархической многоуровневой машины, привилегированном ядре и пользовательском режиме работы транзитных модулей. Модули ядра выполняют базовые функции ОС: управление процессами, памятью, устройствами ввода-вывода и т.п. Ядро составляет сердцевину ОС, без которой она является полностью неработоспособной и не может выполнить ни одну из

своих функций. В ядре решаются внутрисистемные задачи организации вычислительного процесса, недоступные для приложения.



**Рис. 1.3.** Структурированная архитектура

Особый класс функций ядра служит для поддержки приложений, создавая для них так называемую прикладную программную среду. Приложения могут обращаться к ядру с запросами – системными вызовами – для выполнения тех или иных действий, например, открытие и чтение файла, получение системного времени, вывода информации на дисплей и т.д. Функции ядра, которые могут вызываться приложениями, образуют интерфейс прикладного программирования – API (Application Programming Interface).

Для обеспечения высокой скорости работы ОС модули ядра (по крайней мере, большая их часть) являются резидентными и работают в привилегированном режиме (Kernel mode). Этот режим, во-первых, должен обезопасить работу самой ОС от вмешательства приложений, и, во-вторых, должен обеспечить возможность работы модулей ядра с полным набором машинных инструкций, позволяющих собственно ядру выполнять управление ресурсами компьютера, в частности, переключение процессора с задачи на задачу, управлением устройствами ввода-вывода, распределением и защитой памяти и др.

Остальные модули ОС выполняют не столь важные функции, как ядро, и являются транзитными. Например, это могут быть программы архивирования данных, дефрагментации диска, сжатия дисков, очистки дисков и т.п.

Вспомогательные модули обычно подразделяются на группы:

- утилиты – программы, выполняющие отдельные задачи управления и сопровождения вычислительной системы;
- системные обрабатывающие программы – текстовые и графические редакторы (Paint, Imaging в Windows 2000), компиляторы и др.;
- программы предоставления пользователю дополнительных услуг (специальный вариант пользовательского интерфейса, калькулятор, игры, средства мультимедиа Windows 2000);
- библиотеки процедур различного назначения, упрощения разработки приложений, например, библиотека функций ввода-вывода, библиотека математических функций и т.п.

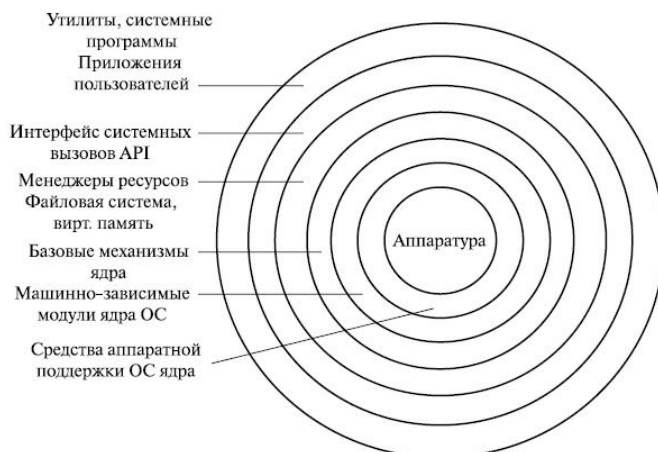
Эти модули ОС оформляются как обычные приложения, обращаются к функциям ядра посредством системных вызовов и выполняются в пользовательском режиме (user mode). В этом режиме запрещается выполнение некоторых команд, которые связаны с функциями ядра ОС (управление ресурсами, распределение и защита памяти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

#### **4. Многослойная структура ОС**

В концепции многоуровневой (многослойной) иерархической машины структура ОС также представляется рядом слоев. При такой организации каждый слой обслуживает вышележащий слой, выполняя для него некоторый набор функций, которые образуют межслойный интерфейс. На основе этих функций следующий верхний по иерархии слой строит свои функции – более сложные и более мощные и т.д. Такая организация системы

существенно упрощает ее разработку, т.к. позволяет сначала "сверху вниз" определить функции слоев и межслойные интерфейсы, а при детальной реализации, двигаясь "снизу вверх", – наращивать мощность функции слоев. Кроме того, модули каждого слоя можно изменять без необходимости изменений в других слоях (но не меняя межслойных интерфейсов!).

Многослойная структура ядра ОС может быть представлена, например, вариантом, показанным на рис. 1.4.



**Рис. 1.4.** Многослойная структура ОС

В данной схеме выделены следующие слои.

1. *Средства аппаратной поддержки ОС.* Значительная часть функций ОС может выполняться аппаратными средствами [10]. Чисто программные ОС сейчас не существуют. Как правило, в современных системах всегда есть средства аппаратной поддержки ОС, которые прямо участвуют в организации вычислительных процессов. К ним относятся: система прерываний, средства поддержки привилегированного режима, средства поддержки виртуальной памяти, системный таймер, средства переключения контекстов процессов (информация о состоянии процесса в момент его приостановки), средства защиты памяти и др.

2. *Машинно-зависимые модули ОС.* Этот слой образует модули, в которых отражается специфика аппаратной платформы компьютера. Назначение этого слоя – "экранирование" вышележащих слоев ОС от особенностей аппаратуры (например, Windows 2000 – это слой HAL (Hardware Abstraction Layer), уровень аппаратных абстракций).

3. *Базовые механизмы ядра.* Этот слой модулей выполняет наиболее примитивные операции ядра: программное переключение контекстов процессов, диспетчерскую систему прерываний, перемещение страниц между основной памятью и диском и т.п. Модули этого слоя не принимают решений о распределении ресурсов, а только обрабатывают решения, принятые модулями вышележащих уровней. Поэтому их часто называют исполнительными механизмами для модулей верхних слоев ОС.

4. *Менеджеры ресурсов.* Модули этого слоя выполняют стратегические задачи по управлению ресурсами вычислительной системы. Это менеджеры (диспетчеры) процессов ввода-вывода, оперативной памяти и файловой системы. Каждый менеджер ведет учет свободных и используемых ресурсов и планирует их распределение в соответствии с запросами приложений.

5. *Интерфейс системных вызовов.* Это верхний слой ядра ОС, взаимодействующий с приложениями и системными утилитами, он образует прикладной программный интерфейс ОС. Функции API, обслуживающие системные вызовы, предоставляют доступ к ресурсам системы в удобной компактной форме, без указания деталей их физического расположения.

### **3.7.3 Результаты и выводы:**

В ходе выполнения практической работы были изучены следующие вопросы

1. Понятие архитектуры операционной системы
2. Монолитная архитектура
3. Структурированная архитектура
4. Многослойная структура ОС

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.

### **3.8 Практическое занятие №8 ( 2 часа).**

**Тема:** «Операционная система Windows XP»

#### **3.8.1 Задание для работы:**

- 1 Ознакомьтесь с элементами Рабочего стола, с главным меню "кнопка Пуск" и панелью задач ОС Windows.
- 2.Работа с окном программ в ОС Windows.
- 3.Работа с окнами программ в ОС Windows.
- 4.Контекстное меню рабочего стола ОС Windows.
- 5.Завершение работы ОС Windows.

#### **3.8.2 Краткое описание проводимого занятия:**

1. Включите ПК

Нажмите кнопку Power на системном блоке ПК.

#### ***2. Ознакомьтесь с элементами Рабочего стола ОС Windows***

После полной загрузки ОС Windows ознакомьтесь с основными пиктограммами (значками и ярлыками), расположенными на рабочем столе:

1. Мои документы;
2. Мой компьютер;
3. Сетевое окружение;
4. Internet Explorer;
5. Корзина.

#### ***3. Ознакомьтесь с главным меню "кнопка Пуск" ОС Windows***

Кнопка Пуск, размещенная на Панели задач, предназначена для вызова Главного меню ОС Windows. Главное меню появляется после щелчка по кнопке Пуск. Главное меню состоит из трех секций разделенных горизонтальными линиями. Установите курсор мыши на команду, справа от которой есть значок ветвления (треугольник), рядом с Главным меню откроется подменю. Ознакомьтесь со стандартными элементами Главного меню.

#### ***4. Ознакомьтесь с Панелью задач в ОС Windows***

На панели задач размещаются программные кнопки, индикаторы на панели индикации и Панель быстрого запуска.

Выполните следующее:

1. Определите, какие программные кнопки отображаются на панели задач, какие программы отображаются на панели быстрого запуска и панели индикации. Для этого подведите указатель мыши к пиктограмме на панели задач - появится всплывающая подсказка с названием программы.
2. Щелчком мыши запустите любую программу из панели быстрого запуска (если на панели быстрого запуска имеются пиктограммы).
3. Измените язык на языковой панели. Для этого щелкните на языковую панель и установите флажок для выбранного языка.

#### ***5. Работа с окном программ в ОС Windows***

Установите чистый флоппи-диск в накопитель для гибких магнитных дисков. Откройте папку Мой компьютер и дважды щелкните на пиктограмму Диск 3,5 (А), откроется окно программ Диск 3,5 (А).

- 5.1 Ознакомьтесь с элементами программного окна (строкой заголовка, системным меню, кнопками управления окном, строкой меню, панель инструментов, адресной строкой, строкой состояния, линейками прокрутки окна) открытого окна Диск 3,5 (А).
- 5.2 Измените положение окна на экране, для чего установите указатель мыши на строку заголовка и нажмите левую кнопку мыши, удерживая нажатой кнопку, выполните перемещение окна.
- 5.3 Измените размеры окна по горизонтали, вертикали и диагонали с помощью манипулятора мышь. Для этого установите указатель мыши на границу окна нажмите левую кнопку мыши, удерживая нажатой кнопку, выполните перемещение границы окна.
- 5.4 Создайте на диске папку Экономическая информатика. Для этого выполните команду Файл/Создать, Папку и введите имя Экономическая информатика, нажмите Enter.
- 5.5 Измените имя папки. Для этого щелкните правой кнопкой мыши на папке Экономическая информатика и в открывшемся контекстном меню выберите команду Переименовать, введите новое имя Эконом\_информатика, нажмите клавишу Enter.
- 5.6 Сверните, Разверните, "Сверните в окно" и закройте окно Диск 3,5 (А) с помощью кнопок в строке заголовка.

#### ***6. Работа с окнами программ в ОС Windows***

Откройте окна дисков А:, С: и папок "Мои документы", "Мои рисунки".

- 6.1 Смените активное окно с помощью программных кнопок или клавишами ALT+TAB.

Выберите требуемое активное окно, которое будет отображаться на экране (остальные окна спрятаны за активным окном), щелкая по программным кнопкам на панели задач. Другой способ смены активного окна - это клавиши ALT+TAB. Нажмите и удерживайте клавишу ALT, а нажимая клавишу TAB, переходите от одного окна к другому.

## 6.2 Измените расположение всех открытых окон.

Для изменения расположения открытых окон воспользуйтесь контекстным меню. Для этого щелкните правой кнопкой мыши на пустое место Панели задач и в открывшемся контекстном меню выберите команду Окна каскадом, Окна сверху вниз или Окна слева направо.

Для отмены выбранного расположения активных окон, щелкните повторно правой кнопкой мыши на пустое место Панели задач и выберите команду Отменить окна рядом или Отменить окна каскадом.

## 6.3 Закройте окна дисков A:, C: и папок "Мои документы", "Мои рисунки".

### ***7. Контекстное меню рабочего стола в ОС Windows***

Выполните упорядочивание значков на рабочем столе. Для этого щелкните правой кнопкой мыши на рабочем столе и в открывшемся контекстном меню выберите команду: Упорядочить значки. В открывшемся подменю выберите одну из команд: Имя, Размер, Тип, Изменен.

### ***8. Завершение работы***

Для завершения работы ПК нажмите на кнопку Пуск и выберите пункт Выключить компьютер, откроется окно "Выключить компьютер", щелкните на кнопке Выключение. Завершается работа ОС Windows, и компьютер готовится к безопасному отключению электропитания. Электропитание отключается автоматически.

### **3.8.3 Результаты и выводы:**

В ходе выполнения практической работы студентам были предложены перечень задач по теме практической работы.

На занятии применялись разнообразные формы работы: фронтальная, групповая, индивидуальная. Фронтальная работа проводилась на этапе актуализации по обобщению и систематизации знаний – ребята отвечали на поставленные мной вопросы. Групповая форма работы использовалась на этапе закрепления темы и решения задач у доски. Индивидуальная форма работы - выполнение практической работы. При затруднении выполнения работы мною оказывалась помощь учащимся.

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.

### **3.9 Практическое занятие №9 ( 2 часа).**

**Тема:** «Calc2003»

#### **3.9.1 Задание для работы:**

1. Формирование представления о работе в табличных процессорах;
2. Развитие навыков по работе с элементами электронных таблиц;

3. Формирование умений организации вычислений, использования мастера функций, мастера диаграмм;

### 3.9.2 Краткое описание проводимого занятия:

Все задания выполнять в одном файле. Каждое задание (кроме 4) следует выполнять на отдельном листе. Имя листа должно соответствовать номеру задания.

#### Простые таблицы. Выполните задание 1:

Создайте таблицу по образцу:

Экзаменационные оценки				
№п/п	ФИО	Математика	Физика	Химия
1.	Астахов Коля	2 3 5 3	3	2 4 4 4
2. 3.	Сергеев Олег		4 5 3	
4.	Никитин Стас			
	Аверьянов Саша			
	Средний балл			

Рассчитайте средний балл по каждому ученику и определите средний балл по каждому предмету.

#### Ввод формул. Выполните задание 2:

Создайте на другом листе документа таблицу 2x33.

Занесите в ячейки первой строки данные: в A1 - X, в B1- Y. В первый столбец занесите значения аргумента X, начиная от -1.5 и заканчивая +1.5 с шагом 0.1.

Во второй столбец вставьте формулу,  $=Xx^2+3*Xx^3$ , где Xx - координаты соседней слева ячейки. Скопируйте эту формулу до конца таблицы.

#### Математические функции. Выполните задание 3:

Создайте на другом рабочем листе таблицу 2x102. В первый столбец занесите координату X от -5 до +5 с шагом 0.1, аналогично тому, как вы делали в предыдущем задании.

Во второй столбец занесите координату Y —сложную функцию от X, например,  $\exp(\sin(Xx))$ .

#### Диаграммы и графики. Выполните задание 4:

1. Выделите в таблице из задания 1 столбцы ФИО и средний и постройте гистограмму «Результаты обучения».

2. Выделите в таблице из задания 1 ячейки с названиями предметов и средний балл по предметам и постройте круговую диаграмму «Качество обучения».

3. Выделите в таблице из задания 2 столбцы A и B и постройте график функции.

4. Аналогичным образом постройте график функции к таблице из задания 3.

#### Ссылки. Выполните задание 5:

Создать таблицу для 10 произвольных изделий, выполнить расчеты и оформить таблицу в соответствии с образцом.



Доходы от продажи изделий				
Товар	Количество	Цена	Доход	Налог
Изделие 1	16	425.0руб.	6 800.0руб.	510.0руб.
Средняя цена		471,8руб.		
Количество изделий		194		
Суммарный доход			99 132,0руб.	
Торговый налог		7,50%		
Сумма налога				7 434,9руб.

#### Смешанные ссылки. Выполните задание 6:

Создать таблицу, рассчитать надбавки, выполнить форматирование. Для расчета использовать одну для всех ячеек формулу со смешанными ссылками.

Надбавки по стажу						
		Надбавка				
	Стаж	5-10 лет	11-15 лет	16-20 лет	21-25 лет	26-30 лет
Разряд	Оклад (руб.)	0,05	0,1	0,15	0,2	0,25
10	600					
11	800					
12	1500					

#### Связь листов. Выполните задание 7:

Создать на новом листе таблицу 2. Ввести значения окладов и надбавок из таблицы «Надбавки по стажу» для всех сотрудников. Рассчитать суммы зарплат.

Таблица 2					
Фамилия	Стаж работы	Разряд	Оклад	Надбавка	Сумма
Андреева	17	11			
Бутаков	12	12			
Горбатов	8	10			

#### Функция IF (ЕСЛИ). Выполните задание 8:

Создать на новом листе таблицу. Вычислите комиссионные менеджеров в зависимости от объема выполненной работы.

Если менеджер не выполнил норму, то ставка комиссионных составляет 5,5%. Если менеджер выполнил или перевыполнил норму, то ставка комиссионных составляет – 7,5%.

Норма продаж – 150 000р.



	А	В	С
1	<b>Норма продаж</b>	<b>\$ 150 000,00</b>	
2	Ставка комиссионных	5,5%	
3	Ставка премиальных	7,5%	
4			
5	<b>Менеджеры</b>	<b>Продажи</b>	<b>Комиссионные</b>
6	Орлов	\$ 155 000,00	\$ 11 625,00
7	Воробьев	\$ 162 023,00	\$ 12 151,73
8	Скворцов	\$ 209 123,00	\$ 15 684,23
9	Синицын	\$ 122 354,00	\$ 6 729,47
10	Грачев	\$ 83 351,00	\$ 4 584,31

### Условное форматирование. Выполните задание 9:

Выполните форматирование различным цветом ячейки значений продаж.

В первой таблице на листе выделите диапазон ячеек В6:В10.

Выполните команду Формат Условное форматирование... В открывшемся окне задайте условия:

меньше 150000 – цвет красный

больше 150000 - цвет зеленый

равно 150000 – цвет желтый

### Подсчет по условию. Выполните задание 10:

Вычислите количество менеджеров, которые имеют объем продаж, меньше, равный и превышающих норму.

Для этого выполните следующие действия.

Подготовьте в столбце Е ячейки для ввода формул.

Перейдите в ячейку Е7. Вызовите Мастер функций, найдите в нем функцию COUNTIF. В поле Диапазон укажите диапазон ячеек В6:В15 (в этом диапазоне ведется поиск значений, превышающих 150 000). В поле Критерий введите ячейку Е6 (в этой ячейке содержится условие, выполнение которого проверяется в диапазоне В6:В15).

С	Д	Е
<b>Комиссионные</b>		<b>Продадо &lt; нормы</b>
\$ 8 240,27		<b>&lt;150 000</b>
\$ 12 151,73		4
\$ 15 684,23		
\$ 6 729,47		
\$ 4 584,31		<b>Продадо &gt;= нормы</b>
\$ 15 364,58		<b>&gt;=150 000</b>
\$ 11 250,00		6
\$ 6 077,50		
\$ 16 509,00		
\$ 12 783,75		

### 3.9.3 Результаты и выводы:

В ходе выполнения практической работы студентам были предложены перечень задач по теме практической работы.

На занятии применялись разнообразные формы работы: фронтальная, групповая, индивидуальная. Фронтальная работа проводилась на этапе актуализации по обобщению и систематизации знаний – ребята отвечали на поставленные мной вопросы. Групповая

форма работы использовалась на этапе закрепления темы и решения задач у доски.

Индивидуальная форма работы - выполнение практической работы. При затруднении выполнения работы мною оказывалась помощь учащимся.

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.

### 3.10 Практическое занятие №10 ( 2 часа).

Тема: «Base 2003»

#### 3.10.1 Задание для работы:

1. Создание базы данных в OpenOffice.org Base
2. Открытие базы данных
3. Создание новой таблицы

#### 3.10.2 Краткое описание проводимого занятия:

##### 1.Создание базы данных в OpenOffice.org Base

Чтобы создать новый файл базы данных, можно пойти двумя путями. Если уже открыт какой-либо документ OpenOffice.org, то достаточно выбрать команду меню **Файл — Создать — База данных**. Второй путь предполагает запуск программы **OpenOffice.org Base** через меню графической оболочки **KDE, Gnome**. Как правило, для этого необходимо открыть главное меню, затем **Офис — «OpenOffice.org Base»**. Во время работы часто возникают вопросы по выполнению тех или иных задач. **OpenOffice.org Base** имеет хорошую справочную систему, которая поможет ответить практически на любой вопрос. Для вызова справки выберете пункт **Вызов справки**.

Создание новой базы данных происходит с помощью мастера и состоит из нескольких этапов. На первом этапе в диалоговом окне предлагается выбрать, создавать новую базу данных или открыть уже существующую (рис. 1).

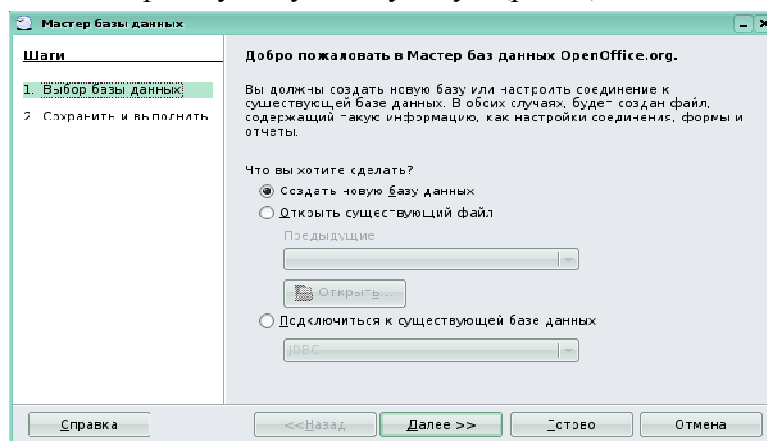


Рис. 1

Чтобы создать новую, необходимо выбрать **Создать новую базу данных** и нажать на кнопку **Далее**.

На втором этапе мастер предлагает два варианта: открыть базу данных для редактирования или сразу приступить к созданию таблиц (рис. 2). После выбора варианта необходимо нажать на кнопку **Готово**.

Чтобы открыть файл базы данных в процессе работы, выберите команду меню **Файл — Открыть**. Из списка **Тип файла** выберите для просмотра только **Документы базы данных**. Выберите документ базы данных и щелкните на кнопку **Открыть**.

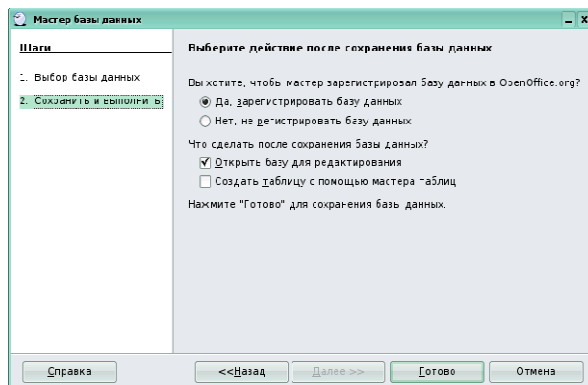


Рис. 2

## 2.Открытие базы данных

Создав базу данных (с нуля, на основе уже существующей базы данных), вы будете периодически открывать ее для внесения изменений или проведения различных операций. Для того чтобы открыть базу данных, выполните следующие действия.

1. Щелкните на кнопке **Открыть** на панели инструментов, и на экране появится предназначенное для этого диалоговое окно (рис. 3).
2. Укажите необходимую папку в списке **Папка** и щелкните на имени файла базы данных, имеющего расширение \* .odb.
3. Щелкните по кнопке **ОК** (рис. 4).

После этого вы можете приступить к формированию таблиц базы данных.

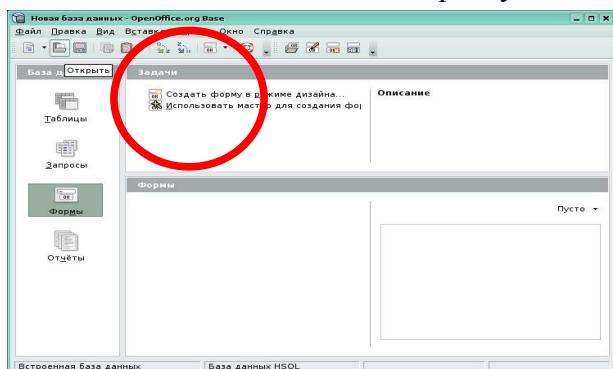


Рис. 3

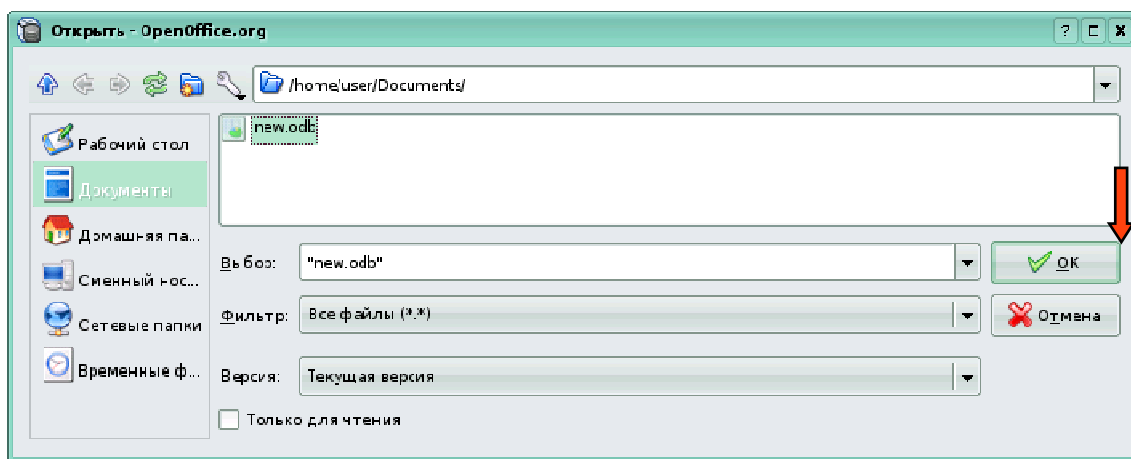


Рис. 4

### 1. Создание новой таблицы

После того как создан новый файл базы данных, редактор откроет основное рабочее окно базы (рис. 5). Здесь можно работать с таблицами, запросами, формами и пр. Создавать, редактировать, управлять ими. Причем практически каждую форму можно создавать в двух режимах — с помощью мастера (для начинающих) или с помощью режима дизайна (для опытных пользователей).

Чтобы воспользоваться помощью мастера таблиц, следует щелкнуть на кнопке **Таблицы** в окне базы данных. В программе OpenOffice.org Base предусмотрено три различных способа создания таблицы базы данных: **Создать таблицу в режиме дизайна** (вся работа по определению структуры таблицы и ее созданию ложится на разработчика), **Использовать мастер для создания таблицы** (позволяет частично автоматизировать процесс создания таблицы), **Создать представление**.

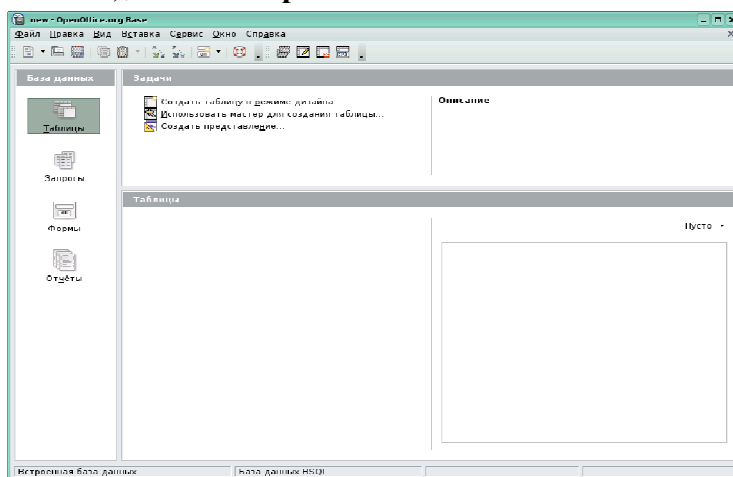


Рис. 5

Рассмотрим создание таблицы с помощью мастера. Для создания таблицы откройте файл базы данных, в которой требуется создать новую таблицу. Щелкните значок **Таблицы**, а справа щелчком левой кнопки мыши выберите **Использовать мастер для создания таблиц...** (рис. 6).

Мастер создания таблиц предлагает множество уже готовых шаблонов. Остается выбрать подходящий и использовать необходимые поля. Создание таблицы начинается с определения ее категории с помощью переключателей **Деловой**, **Персональный**. Затем в

списке **Примеры таблиц** следует выбрать название наиболее подходящего образца. После щелчка по названию выбранного образца формируется соответствующее содержимое списка **Переменные поля**.

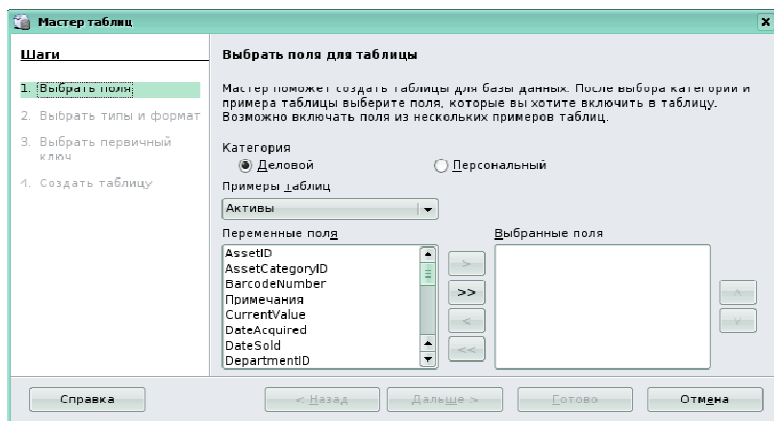



Рис. 6

Следует просмотреть этот список и отобрать те поля, которые предусмотрены в проекте таблицы. Чтобы поле попало в структуру формируемой таблицы, его нужно выделить, а затем щелкнуть по кнопке .

Поле, случайно попавшее в список, можно исключить из него. При желании любое поле, включенное в список выбранных полей, можно переименовать. Это можно сделать в окне следующего шага (рис. 7).

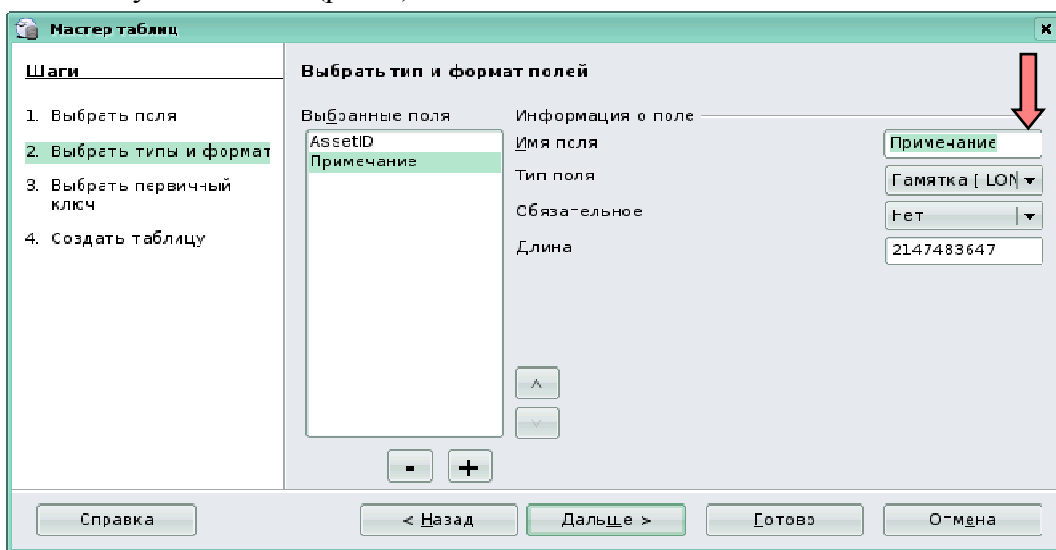


Рис. 7

На третьем шаге происходит задание первичного ключа. В последнем окне мастер предлагает три варианта дальнейших действий: **Немедленно вставить данные**, **Модифицировать дизайн таблицы**, **Создать форму на основе данной таблицы**. Следует сделать выбор и щелкнуть на кнопке **Готово**. В случае выбора первого варианта происходит переключение в режим таблицы, в котором можно выполнить первичное заполнение базы. В случае выбора второго варианта происходит переключение в режим дизайна, в котором можно произвольным образом изменить структуру созданной таблицы.

При выборе третьего варианта мастер таблиц автоматически создаст форму, обеспечивающую более удобный стиль заполнения базы.

Далее рассмотрим процесс создания новой таблицы в режиме дизайна. Чтобы создать новую таблицу в режиме дизайна, выберите **Создание таблицы в режиме дизайна**.

Теперь можно создать поля для таблицы в режиме дизайна. Создадим таблицу «Ученики», содержащую информацию об учениках. Наша таблица будет содержать такие поля: порядковый номер, имя, отчество, фамилия, дата рождения, возраст, класс, параллель, улица, дом, квартира, классный руководитель.

Введем новые поля в строки сверху вниз. Щелкните ячейку **Имя поля** и введите имя для каждого поля данных (рис. 8).

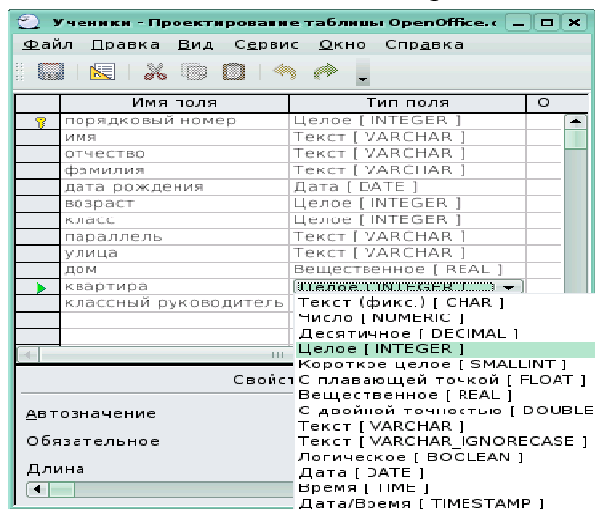


Рис. 8

В следующей ячейке справа определите **Тип поля**. Щелкнув в этой ячейке, выберите тип поля из списка (см. рис. 8).

В нашем примере поля будут иметь следующий тип:

- порядковый номер — целое,
- имя — текст,
- отчество — текст,
- фамилия — текст,
- дата рождения — дата,
- возраст — целое,
- класс — целое,
- параллель — текст,
- улица — текст,
- дом — вещественное,
- квартира — целое,
- классный руководитель — целое (номер).

В каждое поле можно вводить данные, соответствующие заданному типу. Например, нельзя ввести текст в числовое поле. Поля записок в формате dBase III являются ссылками на внутренне управляемые текстовые файлы, которые могут содержать до 64 Кбайт текста.

Для каждого поля можно ввести **Описание**. Текст описания будет возникать в виде всплывающей подсказки при наведении курсора на заголовок в представлении таблицы.

Ниже вводятся **Свойства поля** для каждого выбранного поля данных (рис. 9).

В зависимости от типа базы данных некоторые возможности ввода могут быть недоступны.

В поле **Значение по умолчанию** введите содержимое по умолчанию для каждой новой записи. Это содержимое потом можно будет изменить.

В поле **Обязательное** укажите, может ли поле оставаться пустым.

В поле **Длина** можно открыть список с вариантами выбора. После того как все поля будут заполнены, откройте таблицу для заполнения, для этого в окне базы данных щелкните два раза левой кнопкой мыши на названии таблицы.

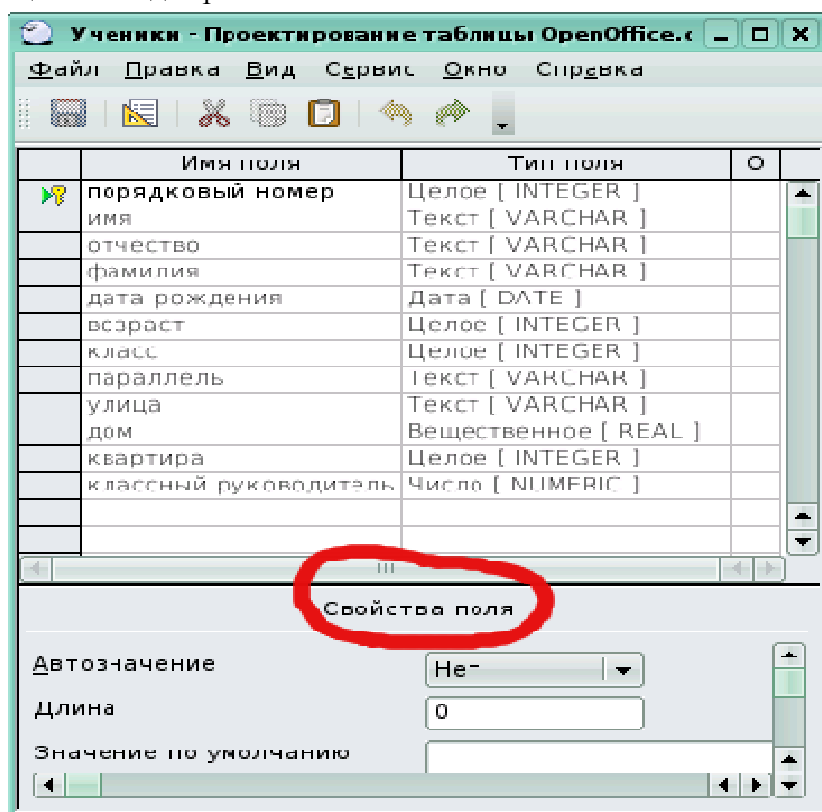


Рис. 9

Заполненная таблица выглядит следующим образом (рис. 10):

порядков...	имя	отчество	фамилия	дата рождения	возраст	класс	парал...	улица	дсм	кварт...	классный руковод
1	Иван	Аркадьевич	Белогов	02.12.96	12	5	а	Ленина	17	114	1
2	Светлана	Андреевна	Сидорова	15.04.96	12	5	б	Парковая	104	26	2
3	Ирина	Ивановна	Аллеева	21.10.96	12	5	а	Садовая	12	68	1

Рис. 10

Записи в таблице можно отсортировать. Для этого:

1. Выделите тот столбец, по которому будет проводиться сортировка.

2. Выберите на панели инструментов кнопку "По возрастанию" или "По убыванию".

В нашей таблице мы отсортировали по алфавиту поле «Фамилия», результат сортировки показан на рис. 11.

порядков...	имя	отчество	фамилия	дата рождения	возраст	класс	парал...	улица	дом	кварт...	классный руковод...
3	Ирина	Ивановна	Аллаева	21.10.96	12	5	а	Садовая	12	68	1
1	Иван	Аркадьевич	Белоногов	02.12.96	12	5	а	Ленина	17	114	1
2	Светлана	Андреевна	Сидорова	15.04.96	12	5	6	Парковая	104	26	2

Рис. 11

### 3.10.3 Результаты и выводы:

В ходе выполнения практической работы студентам были предложены перечень задач по теме практической работы.

На занятии применялись разнообразные формы работы: фронтальная, групповая, индивидуальная. Фронтальная работа проводилась на этапе актуализации по обобщению и систематизации знаний – ребята отвечали на поставленные мной вопросы. Групповая форма работы использовалась на этапе закрепления темы и решения задач у доски. Индивидуальная форма работы - выполнение практической работы. При затруднении выполнения работы мною оказывалась помощь учащимся.

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.

### 3.11 Практическое занятие №11 ( 4 часа).

**Тема:** «Классификация компьютерных сетей. Интернет как единая система ресурсов»

#### 3.11.1 Задание для работы:

1. Теоретические сведения
2. Общий алгоритм, описывающий процесс построения сети.
3. Выполнение работы

#### 3.11.2 Краткое описание проводимого занятия:

##### 1. Теоретические сведения

Процесс построения (проектирования) сети представляет собой упрощенное моделирование не наступившей действительности и включает в себя следующие основные этапы:

1. *Анализ задач*, для решения которых создается сеть, а также определение объема финансирования проекта.
2. *Проектирование физической структуры* - этап, на котором анализируются начальные условия (планировка здания, имеющиеся технические средства и т.п.) и создается детальный проект физической организации сети.



3. *Проектирование инфраструктуры* – этап, на котором определяются протоколы взаимодействия, используемые службы, политика безопасности и т.п. - т.е. логическая организация сети.

4. *Развертывание* - этап, связанный с прокладкой линий связи, установкой и настройкой оборудования.

*Этап анализа* является одним из важнейших, поскольку определяет все остальные решаемые задачи: как физическую структуру сети (например, места расположения компьютеров), так и логическую (используемые протоколы, службы и т.п.). Именно на данном этапе выступает основное различие компьютерных сетей. Основной целью использования учебных компьютерных сетей в образовательных заведениях выступает организационно-методическая поддержка учебно-воспитательного процесса средствами современных сетевых технологий.

На *этапе проектирования* решаются следующие задачи:

1. На основе определенных целевых требований к сети определяется необходимый состав оборудования и, прежде всего, компьютеров: количество, характеристики и т.д.

2. Определяется физическое расположение рабочих мест и определяются этажи и аудитории, которые будут охватываться сетью. При решении этой задачи должна учитываться принципиальная возможность прокладки линий связи к рабочим местам/помещениям.

3. Исходя из решаемых задач, стоимости и расположения, определяется тип физических линий связи, соединяющих рабочие места, состав и расположение коммуникационного оборудования (например, концентраторов).

4. Определяется способ подключения к Интернету: выбирается провайдер – организация, обеспечивающая подключение организации к сети Интернет. При выборе провайдера учитываются факторы: характеристики возможных физических соединений с провайдером, требования к оборудованию и необходимое дополнительное оборудование, начальная стоимость подключения, стоимость эксплуатации подключения, технологические ограничения подключения (невозможность использования некоторых служб).

5. Исходя из технических требований, определяется узел проектируемой сети, который будет являться шлюзом для подключения к Интернету и определяется место его расположения. При этом учитывается удобство физического соединения шлюза с проектируемой сетью и удобство подведения физических линий для подключения к Интернету.

## **2.Приведем общий алгоритм, описывающий процесс построения сети.**

1. Определение исходных данных.
  - Определение целей использования сети.
  - Определение требований к сети
  - Характеристики используемого оборудования (компьютеры, сетевое оборудование, принтеры, модемы и др.)
  - Характеристика сетевого ПО (операционные системы, серверное ПО, антивирусное ПО)
  - Примерная схема здания в котором планируется построить сеть.
2. Проектирование сети
  - Способ сегментирования и объединения сегментов (определение необходимых сегментов оборудования для их формирования).
  - Выбор типа кабеля (как правило выбирается неэкранированная витая пара)
  - Определение активных устройств (модемы, маршрутизаторы и т.п.)
  - Выбор программного обеспечения (серверные и клиентские ОС, серверное программное обеспечение и т.п.).

- Разработка схемы сети (указываются узлы сети и длины соединительных кабелей).

3. Определение стоимости

- Анализ основных направлений затрат
- Составление примерной сметы затрат.

4. Примерный план проведения работ.

5. Развертывание сети.

При создании новой сети желательно учитывать следующие факторы:

- требуемый размер сети (в настоящее время, в ближайшем будущем и по прогнозу на перспективу);
- структура, иерархия и основные части сети (по подразделениям предприятия, а также по комнатам, этажам и зданиям предприятия); основные направления и интенсивность информационных потоков в сети (в настоящее время, в ближайшем будущем и в дальней перспективе); характер передаваемой по сети информации;
- технические характеристики оборудования (компьютеров, адаптеров, кабелей, репитеров, концентраторов, коммутаторов);
- возможности прокладки кабельной системы в помещениях и между ними, а также меры обеспечения целостности кабеля;
- обслуживание сети и контроль ее безотказности и безопасности;
- требования к программным средствам по допустимому размеру сети, скорости, гибкости, разграничению прав доступа, стоимости, по возможностям контроля обмена информацией и т.д. (например, если предполагается использование одного ресурса многими пользователями, то следует использовать серверную ОС);
- необходимость подключения к другим сетям (например, глобальным);
- имеющиеся компьютеры и их программное обеспечение, а также периферийные устройства (принтеры, сканеры и т.д.).

При выборе размера (под размером сети в данном случае понимается как количество объединяемых в сеть компьютеров, так и расстояния между ними) и структуры сети необходимо учитывать:

- количество компьютеров (следует оставлять возможность для дальнейшего роста количества компьютеров в сети);
- требуемую длину линий связи сети (например, если расстояния очень большие, может понадобиться использование дорогого оборудования).
- способы объединения частей сети (для объединения частей сети могут использоваться репитеры, репитерные концентраторы, коммутаторы, мосты и маршрутизаторы, причем в ряде случаев стоимость этого объединительного оборудования может даже превысить стоимость компьютеров, сетевых адаптеров и кабеля.
- Возможность масштабирования (например, лучше приобретать коммутаторы или маршрутизаторы с количеством портов, несколько большим, чем требуется в настоящий момент).

**Пример.** Пусть небольшое предприятие занимает три этажа, на каждом по пять комнат, и включает в себя три подразделения, по три группы. В этом случае можно построить сеть таким образом (рис. 1):

- Рабочие группы занимают по 1–3 комнаты, их компьютеры объединены между собой репитерными концентраторами. Концентратор может использоваться один на комнату, один на группу или один на весь этаж. Концентратор целесообразно расположить в помещении, в которое имеет доступ минимальное количество сотрудников.
- Подразделения занимают отдельный этаж. Все три сети рабочих групп каждого подразделения объединяются коммутатором, а для связи с сетями других подразделений используется маршрутизатор. Коммутатор вместе с одним из концентраторов лучше поместить в отдельной комнате.

- Общая сеть предприятия включает три сегмента сетей подразделений, объединенных маршрутизатором. Этот же маршрутизатор может использоваться для подключения к глобальной сети.
- Серверы рабочих групп располагаются в комнатах рабочих групп, серверы подразделений – на этажах подразделений.

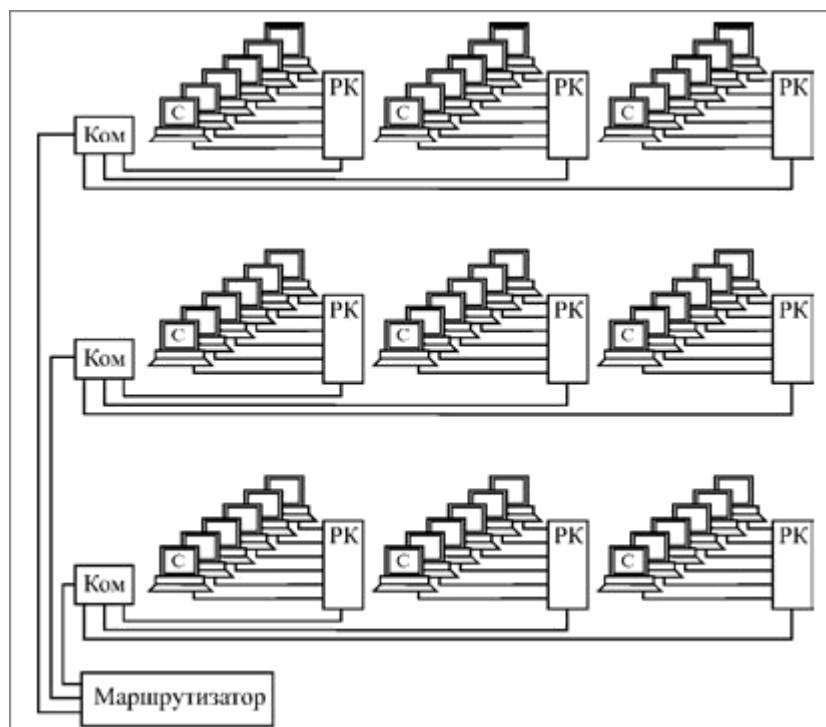


Рисунок 1. Структура сети предприятия (С – серверы рабочих групп, ПК – репитерные концентраторы, Ком – коммутаторы)

При выборе сетевого оборудования надо учитывать множество факторов, в частности:

- уровень стандартизации оборудования и его совместимость с наиболее распространенными программными средствами;
- скорость передачи информации и возможность ее дальнейшего увеличения;
- возможные топологии сети и их комбинации (шина, пассивная звезда, пассивное дерево);
- метод управления обменом в сети (CSMA/CD, полный дуплекс или маркерный метод);
- разрешенные типы кабеля сети, максимальную его длину, защищенность от помех;
- стоимость и технические характеристики конкретных аппаратных средств (сетевых адаптеров, трансиверов, репитеров, концентраторов, коммутаторов).

В настоящее время для организации локальных сетей в подавляющем большинстве случаев используется неэкранированная витая пара UTP. Более дорогие варианты на основе экранированной витой пары, оптоволоконного кабеля или беспроводных соединений применяются на предприятиях, где в этом существует действительно острая необходимость. Например, оптоволокно может использоваться для связи между удаленными сегментами сети без потери скорости.

При выборе сетевого программного обеспечения (ПО) надо, в первую очередь, учитывать следующие факторы:

- Какую сеть поддерживает сетевое ПО: одноранговую, сеть на основе сервера или оба этих типа;

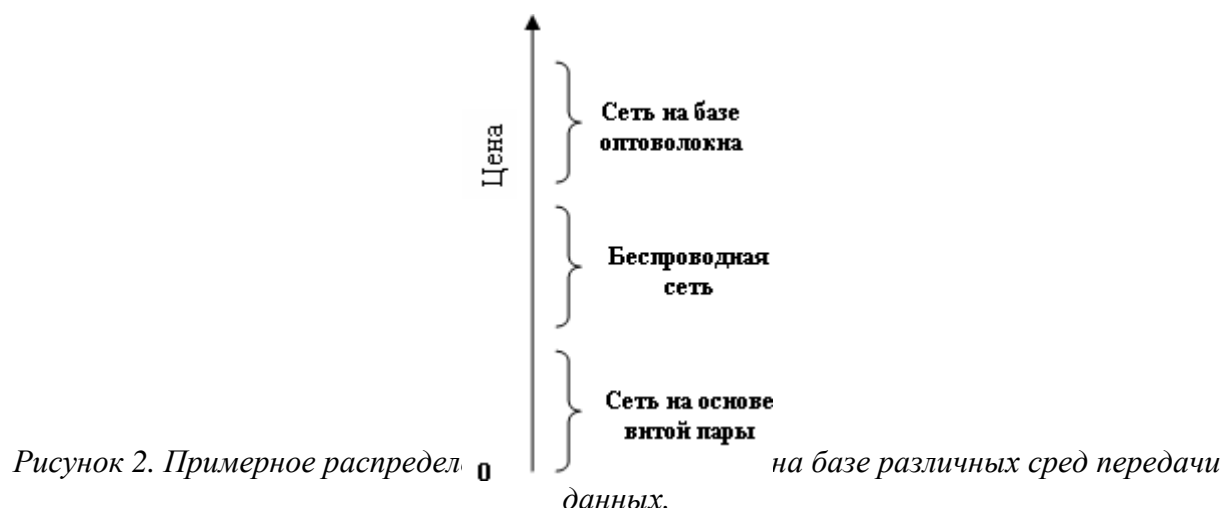
- Максимальное количество пользователей (лучше брать с запасом не менее 20%);
- Количество серверов и возможные их типы;
- Совместимость с разными операционными системами и компьютерами, а также с другими сетевыми средствами;
- Уровень производительности программных средств в различных режимах работы;
- Степень надежности работы, разрешенные режимы доступа и степень защиты данных;
- Какие сетевые службы поддерживаются;
- Стоимость программного обеспечения, его эксплуатации и модернизации.

Еще до установки сети необходимо решить вопрос об управлении сетью. Даже в случае одноранговой сети лучше выделить для этого отдельного специалиста (администратора), который будет иметь всю информацию о конфигурации сети и распределении ресурсов и следить за корректным использованием сети всеми пользователями. Если сеть большая, то одним сетевым администратором уже не обойтись, нужна группа, возглавляемая системным администратором. *После установки и запуска сети решать эти вопросы, как правило, слишком поздно.*

При проектировании следует определить возможные направления финансовых затрат (к данному этапу проектирования необходимые предпосылки для решения этой задачи уже имеются):

- Дополнительные компьютеры и апгрейд существующих компьютеров. Необязательное направление затрат: при достаточном количестве и качестве существующих компьютеров их апгрейд не требуется (или требуется в минимальном объеме – например, для установки более современных сетевых карт); в одноранговой сети не нужен (хотя и желателен) также специальный файл-сервер.
- Сетевые аппаратные средства (кабели и все, что необходимо для организации кабельной системы, сетевые принтеры, активные сетевые устройства – повторители, концентраторы, маршрутизаторы и т.д.).
- Сетевые программные средства, прежде всего, сетевая ОС на необходимое число рабочих станций (с запасом).
- Оплата работы приглашенных специалистов при организации кабельной системы, установке и настройке сетевой ОС, при проведении периодической профилактики и срочного ремонта. Необязательное направление затрат: для небольших сетей со многими из этих работ может и должен справляться штатный сетевой администратор (возможно, с помощью других сотрудников данного предприятия).

Примерное распределение стоимости установки сети с использованием различных сред передачи данных приведено на рисунке 2.



### 3.Выполнение работы

**Задание 1. Учитывая исходную информацию (примерный план здания образовательного заведения, количество и специфику устанавливаемых ПК и где) спроектировать учебную компьютерную сеть (собрать исходные данные; выбрать: размер и структуру сети, оборудование, сетевые программные средства; спроектировать кабельную систему; рассчитать примерную стоимость оборудования).**

1. Ознакомьтесь с решением поставленной задачи:
  - для этого откройте файл, содержащий пример выполнения задания (скачать в формате: [ODT](#), [PDF](#), [DOC](#))
  - ознакомьтесь с содержанием файла;
2. Выполните расчеты стоимости программного обеспечения в таблице «Составление сметы примерных затрат».
3. Модифицируйте план проектирования сети, увеличив количество компьютеров до 36 (например, добавьте еще 1 кабинет информатики на 1-м этаже).
4. Сохраните результат работы в личной папке.

#### 3.11.3 Результаты и выводы:

В ходе выполнения практической работы студентам были предложены перечень задач по теме практической работы.

На занятии применялись разнообразные формы работы: фронтальная, групповая, индивидуальная. Фронтальная работа проводилась на этапе актуализации по обобщению и систематизации знаний – ребята отвечали на поставленные мной вопросы. Групповая форма работы использовалась на этапе закрепления темы и решения задач у доски. Индивидуальная форма работы - выполнение практической работы. При затруднении выполнения работы мною оказывалась помощь учащимся.

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.

### 3.12 Практическое занятие №12 ( 6 часов).

**Тема:** «Создание почтового ящика»

#### 3.12.1 Задание для работы:

1. Теоретические сведения
2. Регистрация на бесплатном почтовом сервере
3. Знакомство с основными возможностями и элементами интерфейса Web-mail
4. Работа с почтовыми сообщениями
5. Заполнение адресной книги

#### 3.12.2 Краткое описание проводимого занятия:

##### Теоретические сведения

*Электронная почта* – одна из наиболее распространенных и популярных функций компьютерных сетей, обеспечивающая обмен сообщениями между пользователями сети.

Порядок использования электронной почты во многом сходен с обычной почтой. Роль почтовых отделений играют узлы сети Интернет – *почтовые серверы*, на которых абонентам организуются специальные *почтовые ящики*.

При пересылке сообщений по электронной почте необходимо указывать адрес получателя в сети Интернет. Он состоит из: имени пользователя, символа @, имени почтового сервера.

Например: sasha\_007@mail.ru

По электронной почте можно пересылать не только текстовые сообщения, но и готовые файлы, созданные в любых других программах.

Работать с электронной почтой можно при помощи почтовой программы (почтового клиента), установленной на компьютере пользователя или при помощи браузера, с помощью web-интерфейса.

*Почтовая программа* (клиент электронной почты, почтовый клиент) — программное обеспечение, устанавливаемое на компьютере пользователя, предназначенное для получения, написания, отправки, хранения и обработки сообщений электронной почты пользователя (например, Microsoft Outlook Express, The Bat!, Netscape Messenger, Mozilla).

В системе пересылки электронной почты еще необходим почтовый сервер (сервер электронной почты). *Почтовый сервер* - это компьютерная программа, которая передаёт сообщения от одного компьютера к другому. Почтовые серверы работают на узловых компьютерах Интернета, а почтовые клиенты должны быть у каждого пользователя e-mail.

Существует большое количество WWW-серверов, которые предлагают завести бесплатный почтовый ящик и позволяют работать с почтой, используя только браузер. Чтобы получить бесплатный почтовый ящик на таком сервере, необходимо зарегистрироваться. Для этого нужно заполнить несколько обязательных полей – ввести свой логин, пароль, возраст, пол и т.д. В случае успешной регистрации, за Вами будет закреплён бесплатный почтовый электронный адрес.

*Спам* – рассылка коммерческой, политической и иной рекламы или иного вида сообщений лицам, не выразившим желания их получать. Старайтесь не рассылать одно письмо сразу большому количеству людей, т.к. многие могут воспринять это письмо как спам (нежелательную корреспонденцию).

*Спамер* – пользователь, рассылающий спам по интернету, локальным сетям, системам сотовой связи, и т. д.

##### Технология выполнения задания:

**Задание 1.** Регистрация на бесплатном почтовом сервере.

Зарегистрироваться на одном из бесплатных серверов [www.yandex.ru](http://www.yandex.ru), [www.mail.ru](http://www.mail.ru), [www.nm.ru](http://www.nm.ru), [www.rambler.ru](http://www.rambler.ru), [www.ok.ru](http://www.ok.ru), [www.pochta.ru](http://www.pochta.ru) и т.п.

1. Запустите интернет-браузер **Internet Explorer** или **Opera** с помощью значка на **Рабочем столе**.

2. В адресной строке браузера введите адрес сайта (например, [www.yandex.ru](http://www.yandex.ru)).

**Яндекс**  
Найдётся всё

3. Выберите ссылку **Почта - Зарегистрироваться** или **Завести почтовый ящик**.

4. Заполните форму регистрации.

*Примечание.* Помните, что

- при введении **Вашего имени** и **Фамилии** будут предложены автоматически свободные логины, понравившийся вы можете выбрать или придумать собственный, который будет проверен почтовым сервером, занят ли он другим пользователем.
- поля **Логин**, **Пароль** и **Подтверждение пароля** должны заполняться латинскими буквами, причем пароль должен содержать не менее 4-х символов;
- обязательные поля для заполнения отмечены звездочками.

5. Подтвердите данные, нажав кнопку **Зарегистрировать**.

6. После успешной регистрации появляется ваш личный адрес.

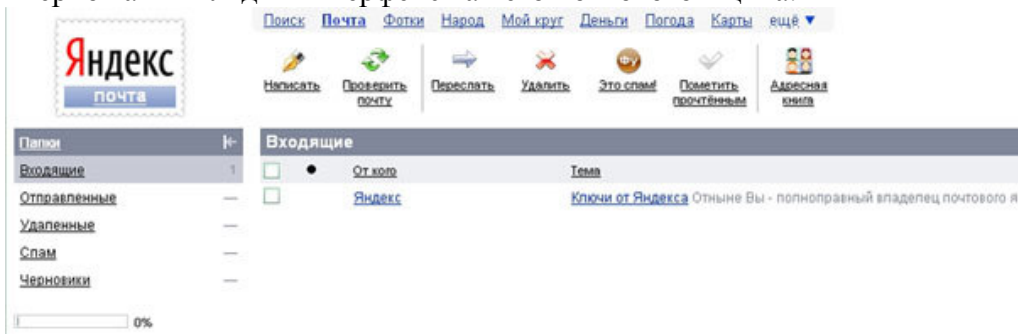
7. Подтвердите согласие, нажав кнопку **Сохранить**.

**Задание 2.** Знакомство с основными возможностями и элементами интерфейса Web-mail.

1. Откройте свой новый почтовый ящик на бесплатном почтовом сервере и изучите основные элементы интерфейса.

Логин:   
Пароль:  [Забыли пароль?](#)

Примерно так выглядит интерфейс вашего почтового ящика:



*Примечание:*

- Папка **Входящие** содержит всю поступившую к вам корреспонденцию (на ваш почтовый ящик).
- Папка **Отправленные** содержит всю отправленную вами другим адресатам в Internet корреспонденцию.
- В папку **Рассылки** складываются письма, которые были одновременно разосланы большому числу пользователей.
- Папка **Удаленные** хранит удаленные письма из любой другой папки.
- Папка **Черновики** хранит не отправленные письма.

**Задание 3.** Работа с почтовыми сообщениями.

1. Создайте сообщение с темой «**ФИО**»:

- щелкните по кнопке **написать**;

заполните заголовки сообщения: **Кому**, **Копия**, **Тема** следующим образом: в заголовке **Кому** укажите адрес преподавателя [kashaev\\_1971@mail.ru](mailto:kashaev_1971@mail.ru), **Копия** – адрес соседа справа. В качестве **Темы** укажите «**ФИО**»;

- впишите свои фамилию, имя, отчество, номер группы в текст сообщения.

2. Отправьте сообщение с помощью кнопки **Отправить**.

3. Перейдите в папку **Входящие**. Вам должно прийти сообщение от соседа слева. Для того, чтобы прочитать полученное сообщение, необходимо нажать на ссылку в поле **От кого**.

4. В появившемся окне нажмите на кнопку **Ответить**. Напишите ответ на это письмо и нажмите на кнопку **Отправить**.



5. Создайте новое сообщение и **вложите в него текстовый файл**:
  - На рабочем столе правой кнопкой мыши создайте документ **Microsoft Word**, назовите «Приглашение», наберите текст приглашения на день рождения, закройте файл, сохраните;
  - вернитесь в свой электронный ящик;
  - щелкните по кнопке **Написать**.
  - заполните заголовки сообщения: **Кому, Копия, Тема** следующим образом: в заголовке **Кому** укажите адрес соседа справа. В качестве **Темы** укажите «**Приглашение**»;
  - нажмите на кнопку **Обзор**, укажите местонахождение файла (**Рабочий стол**);
  - напишите текст сообщения.
6. Отправьте сообщение, нажав на соответствующую кнопку.
7. Создайте новое сообщение и **вложите в него графический файл**:
  - заполните заголовки сообщения: **Кому, Копия, Тема** следующим образом: в заголовке **Кому** укажите адрес соседа справа. В качестве **Темы** укажите «**Картинка**»;
  - нажмите на кнопку **Обзор**, укажите местонахождение файла (свою папку **Общие документы/181/...**);
  - напишите текст сообщения.
8. Отправьте сообщение, нажав на соответствующую кнопку.
9. Перейдите в папку **Входящие**. В списке сообщений найдите электронное письмо с темой «**Приглашение**», отправленное соседом слева. Значок в виде скрепки свидетельствует о наличии в полученном письме вложения. Сохраните вложенный файл в свою папку **Общие документы/181/...**
  - откройте полученное сообщение;
  - щелкните по значку вложенного файла левой кнопкой мыши;
  - в появившемся окне нажмите на кнопку **Сохранить**;
  - укажите путь сохранения
10. Сообщение с темой «**Приглашение**» перешлите преподавателю:
  - откройте нужное письмо и нажмите на кнопку **Переслать**;заполните поле **Кому**, впишите электронный адрес преподавателя [kashaev\\_1971@mail.ru](mailto:kashaev_1971@mail.ru) и отправьте сообщение.

**Задание 4.** Заполнение адресной книги.

Занесите в Адресную книгу новых абонентов.

1. Пополните **Адресную книгу**, воспользовавшись пунктом меню **Сервис - Адресная книга** или соответствующей кнопкой на панели инструментов.
2. Внесите в **Адресную книгу** преподавателя, соседа справа и слева. Для этого выполните команду **Файл - Создать контакт** (или щелкните левой кнопкой мыши на кнопке **Создать** и выберите пункт меню **Создать контакт**). Внимательно изучите вкладки, представленные в данном диалоговом окне. Обратите внимание на то, что в нем имеются средства для ввода как личной, так и служебной информации (для практической деятельности, как правило, достаточно заполнить лишь несколько полей на вкладке **Имя**).
3. Начните заполнение полей вкладки **Имя** с поля **Имя в книге**. Введите сюда такую запись, которую хотели бы видеть в списке контактов, например Сорокин И.И.;
4. Заполните поля **Фамилия** (Сорокин), **Имя** (Иван) и **Отчество** (Иванович);
5. В поле **Адреса электронной почты** введите его электронный адрес.
6. Занесите введенные данные в **Адресную книгу**, нажав на кнопку **Добавить**.

*Примечание.* Если необходимо изменить внесенные данные, следует щелкнуть на записи правой кнопкой мыши, в контекстном меню выбрать пункт **Свойства** и перейти на вкладку **Имя**.

### 3.12.3 Результаты и выводы:

В ходе выполнения практической работы студентам были предложены перечень задач по теме практической работы.



На занятии применялись разнообразные формы работы: фронтальная, групповая, индивидуальная. Фронтальная работа проводилась на этапе актуализации по обобщению и систематизации знаний – ребята отвечали на поставленные мной вопросы. Групповая форма работы использовалась на этапе закрепления темы и решения задач у доски. Индивидуальная форма работы - выполнение практической работы. При затруднении выполнения работы мною оказывалась помощь учащимся.

Были сформированы навыки самостоятельной работы, систематизация полученных знаний. Студенты научились делать выводы, научились преодолевать трудности для достижения намеченной цели.