

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ  
ГОУ ВПО  
«АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ  
Зав. кафедрой ОМиИ  
\_\_\_\_\_ Г.В. Литовка  
«\_\_\_\_\_» \_\_\_\_\_ 2007 г.

# ИНФОРМАТИКА

## УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

*для специальностей:*

*040101 – Социальная работа*

*040201 – Социология*

Составители: А.Н. Киселева, старший преподаватель

О.В. Ефимова, ассистент

Т.А. Макачук, к.п.н., доцент

Н.А. Чалкина, к.п.н., доцент

Благовещенск, 2007

*Печатается по решению  
редакционно-издательского совета  
факультета математики и информатики  
Амурского государственного университета*

***Киселева А.Н., Ефилова О.В., Макачук Т.А., Чалкина Н.А.***

**Учебно-методический комплекс дисциплины «Информатика» для специальностей  
040201, 040101. – Благовещенск: АмГУ, 2007. – 282 с.**

© Амурский государственный университет, 2007

© Кафедра общей математики и информатики, 2007

## СОДЕРЖАНИЕ

I. Рабочая программа.....	4
1. Цели и задачи дисциплины, ее место в учебном процессе.....	4
2. Содержание учебной дисциплины «Информатика».....	4
2.1. Стандарт (по предмету).....	4
2.2. Темы дисциплины и их содержание.....	5
2.3. Тематическое планирование лабораторных занятий.....	6
2.4. Распределение учебного времени (объем в часах).....	8
2.5. Вопросы для самостоятельной работы.....	13
3. Перечень промежуточных форм контроля знаний студентов.....	13
3.1. Вопросы к экзамену для специальности 040101 .....	14
3.2. Вопросы к зачету для специальности 040201.....	15
3.3. Вопросы к экзамену для специальности 040201 .....	16
4. Учебно-методические материалы.....	19
4.1. Основная литература.....	19
4.2. Дополнительная литература.....	19
II. Методические рекомендации профессорско-преподавательскому составу.....	20
1. Методические рекомендации по проведению лекционных занятий.....	20
2. Методические рекомендации по проведению лабораторных занятий.....	20
3. Методические рекомендации по организации контроля знаний студентов .....	21
III. Конспекты лекций.....	22
IV. Методические указания по выполнению лабораторных работ.....	95
V. Методические указания по выполнению домашних заданий.....	95
VI. Перечень программных продуктов, используемых при преподавании дисциплины.....	96
VII. Комплект заданий для лабораторных работ.....	97
VIII. Комплект заданий для контрольных работ.....	246
IX. Задания для зачета.....	274
X. Примерные экзаменационных билетов для специальности 040101 .....	280
XI. Примерные экзаменационных билетов для специальности 040201.....	281
XII. Карта обеспеченности дисциплины кадрами профессорско- преподавательского состава.....	282

## **I. РАБОЧАЯ ПРОГРАММА**

**1. Цели и задачи учебной дисциплины «Информатика», ее место в учебном процессе.**

### **1.1. Место дисциплины в учебном процессе**

Предлагаемый курс обеспечивает базовую подготовку студентов в области использования средств вычислительной техники: для всех курсов, использующих автоматизированные методы анализа и расчетов, курсового и дипломного проектирование.

### **1.2. Цели преподавания учебной дисциплины «Информатика»:**

- формирование личности студента, развитие его интеллекта и способностей к логическому мышлению;
- обучение основным приемам работы на ЭВМ.

### **1.3. Задачи изучения дисциплины:**

- выработать умения анализировать полученные результаты, привить навыки самостоятельного изучения литературы по информатике
- научить студентов приемам работы с программными средствами реализации информационных процессов.

### **1.4. По завершению обучения по дисциплине студент должен:**

- знать и уметь выполнять все основные манипуляции, связанные с мышью;
- уметь правильно пользоваться программными средствами;
- знать основы и возможности современной компьютерной технологии;
- уметь работать в условиях возможного заражения ПК вирусами;
- работать с программными средствами обслуживающего назначения;
- быстро и точно выполнять операции копирования,
- переименования, перемещения, удаления файлов и каталогов, ярлыков и папок.
- знать способы защиты информации;
- иметь понятие о современных информационных сетях;
- быстро и качественно выполнять поиск информации в сети;
- основные типы алгоритмов;
- языки программирования и стандартные программные обеспечения своей профессиональной деятельности;

**2. Содержание учебной дисциплины «Информационные технологии в социально-культурном сервисе и туризме».**

### **2.1. Стандарт по предмету для специальности 040101**

Понятие информации, общая характеристика процессов сбора, передачи, обработки и накопления информации: технические и программные средства реализации

информационных процессов, модели решения функциональных и вычислительных задач, алгоритмизация и программирования; языки программирования высокого уровня; базы данных, программное обеспечение и технологии программирования; локальные и глобальные сети ЭВМ. Компьютерный практикум.

### **Стандарт по предмету для специальности 040201**

Понятие информации, общая характеристика процессов сбора, передачи, обработки и накопления информации: технические и программные средства реализации информационных процессов, модели решения функциональных и вычислительных задач, алгоритмизация и программирование; языки программирования высокого уровня; базы данных, программное обеспечение и технологии программирования; локальные и глобальные сети ЭВМ; основы защиты информации и сведений, составляющих государственную тайну; методы защиты информации. Компьютерный практикум.

#### **2. 2. Темы дисциплины и их содержание.**

##### **1. Общие характеристики процессов сбора, передачи, обработки и накопления информации.**

Понятие информации. Знания и данные. Форма представления информации. Виды информации. Свойства информации. Позиционные системы счисления информации. Общая структурная схема информационного процесса. Информационные системы и технологии. Информационное общество. Понятие системы счисления. Классификация систем счисления. Двоичная система счисления. Правила перевода из десятичной в двоичную систему. Правила перевода из двоичной в десятичную систему счисления. Системы счисления, родственные двоичной (восьмеричная, шестнадцатеричная).

##### **2. Технические средства реализации информационного процесса.**

История развития вычислительной техники. Классы ЭВМ и их основные характеристики. Основные блоки ПК и их назначение. Процессор и его характеристики: разрядность, тактовая частота, быстродействие. Запоминающие устройства. Периферийные устройства: монитор, клавиатура, принтер, сканер, модем, графопостроитель, дигитайзер, манипуляторы, средства мультимедиа.

##### **3. Программные средства реализации информационных процессов.**

Классификация программных средств. Операционная система: понятие, составные части, классификация. Операционные оболочки. Сервисные программные средства: форматирование, дефрагментация, проверка диска, очистка диска, сведения о системе. Архивация данных. Краткий обзор современных программных средств.

##### **4. Локальные сети ЭВМ.**

Понятие компьютерной сети. Устройства сети: сервер, рабочая станция, коммуникационные узлы. Характеристики сети. Классификация сетей по территориальному признаку. Топология компьютерных сетей. Линии связи. Стек протоколов TCP/IP.

## **5. Глобальные компьютерные сети.**

Понятие Интернет. Современная структура сети Интернет. Адресация в сети Интернет: IP, URL. Основные протоколы сети Интернет. Электронная почта. Word Wide Web: URL-адрес, гипертекст, программы-браузеры, язык разметки гипертекста HTML. Поиск информации в Интернет. UseNet, Chat, IP-телефония.

## **6. Защита информации в сети.**

Основы защиты информации и сведений, составляющих государственную тайну: угрозы информации в сети, основные аспекты безопасности. Методы защиты информации: криптография, электронная подпись, аутентификация, сертификация Web-узлов. Компьютерные вирусы и антивирусные программные средства.

## **7. Компьютерная графика.**

Растровая, векторная, фрактальная графика. Форматы графических файлов: bmp, gif, jpeg, png, ppt, tiff, wmf, sda. Интерфейс графического редактора: CorelDraw. Компьютерные презентации, Power Point.

## **8. Базы данных.**

Понятие базы данных. Модели организации данных. Язык SQL. Системы управления базами данных. Основные понятия СУБД Access: поле данных, ключ поля данных, схема данных, таблицы, формы, запросы, отчеты.

## **9. Основы программирования.**

Алгоритмизация и программирование. Методика подготовки и решения задачи на ПК. Основные понятия теории алгоритмов. Способы записи алгоритма. Алгоритм и его свойства. Языки программирования высокого уровня.

### **2.3. Тематическое планирование лабораторных занятий по курсу "Информатика". Компьютерный практикум.**

#### **1. Операционная система Windows.**

Папка "Мой компьютер". Работа с файлами, папками, ярлыками: создание, открытие, переименование, копирование, перемещение, удаление. Работа с деревом каталогов (файловой структурой) в программе "Проводник". Восстановление файлов с помощью программы "Корзина". Виды меню: системное, строка меню, контекстное, главное, панель инструментов. Элементы интерфейса: радиокнопки, флажки, командные кнопки. Основы работы со стандартными программами Windows: Блокнот, калькулятор, графический редактор Paint. Панель управления. Поиск файлов и папок. Настройка интерфейса Windows, в том числе настройка главного меню.

#### **2. Операционные оболочки, Total Commander.**

Работа с панелями информационного окна, управление пакетом с помощью функциональных клавиш и ниспадающего меню. Создание, открытие, просмотр,

копирование, перемещение, переименование, удаление файлов и каталогов. Работа с архивными файлами.

### **3. Сервисное программное обеспечение Windows.**

Программы обслуживания дисков, сведения о системе, программы-архиваторы (WinZip, WinRar), антивирусные программы.

### **4. Текстовый процессор Word.**

Классификация текстовых редакторов: простейшие текстовые редакторы, текстовые процессоры, издательские системы. Параметры страницы. Форматирование абзаца, опции шрифта, создание списка. Колонтитулы. Вставка объектов (Equation, WordArt, ClipArt и др.). Создание ссылок (концевые сноски и оглавление). Оформление таблиц и вычисления в них. Проверка орфографии.

### **5. Электронная таблица Excel.**

Назначение Excel. Адресация ячеек. Форматирование ячеек. Создание и копирование формул. Математические, статистические и логические функции. Сортировка и фильтрация данных. Графический анализ данных: гистограмма, линейный график, круговая диаграмма.

### **6. Система управления базами данных ACCESS.**

Построение структура реляционной базы данных: определение полей, ключевого поля, тип и формат полей. Режим конструктора. Объекты БД: таблица, форма, запрос. Сортировка и фильтрация данных. Схема данных: связи между таблицами  $1 \leftrightarrow 1$ ,  $1 \leftrightarrow \infty$ ,  $\infty \leftrightarrow \infty$ . Построение простых запросов.

### **7. Компьютерная графика CorelDRAW.**

Создание графических изображений, системы цветов и форматов графических файлов. Создания рекламы, коллажей и многого другого. Работа со слоями и цветокоррекция изображений, работа с каналами и масками.

### **8. Презентации PowerPoint**

Рисунки и графические примитивы на слайдах. Выбор дизайна презентации. Редактирование и сортировка слайдов. Использование анимации в презентации. Интерактивная презентация – переходы между слайдами. Демонстрация презентации.

### **9. Знакомство с Delphi for Object Pascal**

Интерфейс Delphi: окно формы, окно Инспектора Объектов, окно кода программы, панель компонентов. Компоненты TPanel, TButton. Задание свойств и событий. Описание реакции на событие. Программа изменение цвета надписи по щелчку мыши на соответствующей командной кнопке.

### **10. Технология программирования задач линейной структуры**

Компоненты TLabel, TEdit, TPanel. Функции перевода строки в число, числа в строку. Составление программы вычисления значения выражения, типа: .

### **11. Технология программирования задач разветвляющейся структуры**

Компоненты TMemo, TCheckBox. Программирование задачи нахождения максимального (минимального) из чисел, например:  $\max(x+y+z, x*y*z, 3)$ . Написать программу, вычисляющую  $f(x)$  на промежутке.

### 12. Технология программирования задач с множественным выбором.

Компоненты TListBox – список строк, T radiogroup – список радиокнопок. Оператор выбора, константы выбора. Программы: по выбранному элементу списка выдается соответствующая информация; по введенному значению выделяется соответствующая строка списка.

### 13. Технология программирования задач циклической структуры

Циклы с параметром, с предусловием, с постусловием. Составление программ вычисления суммы, произведения, среднего арифметическое членов числового ряда.

### 14. Технология программирования задач с данными типа одномерный массив (вектор)

Описание данных типа одномерный массив. Написание программ, в которых для одномерного массива осуществляется: задание элементов массива (псевдослучайно, по определенному правилу, пользователем); нахождение суммы (произведения) элементов массива; определение количества элементов, соответствующих заданному условию; нахождение максимального (минимального) элемента массива; сортировка элементов вектора.

### 15. Технология программирования задач с данными типа матрица

Описание данных типа двумерный массив. Написание программ, в которых для двумерного массива осуществляется: нахождение суммы (произведения) строк, столбцов матрицы; определение количества элементов n-строки, соответствующих заданному условию; нахождение максимального (минимального) элемента n-столбца матрицы; получение новой матрицы.

## 2.4. Распределение учебного времени (в часах).

№	Тема занятия	Лек
	<i>1 семестр (лекции- 18ч.)</i>	
1	Информатизация общества. Роль информатизации в развитии общества. Информатика как наука: предмет и задачи. Появление и развитие информатики. Структура информатики. Данные. Информация. Информация в материальном мире. Свойства информации. Передача информации. Общая характеристика процессов сбора, передачи, обработки и накопления информации	2 ч.
2	Структурная и функциональная организация ЭВМ. Микропроцессор, сопроцессор, память, контроллеры и шина, электронные платы, накопители. Последовательность прохождения информации при ее обработке.	2 ч.
3	Основные внешние устройства ПК. Классификация ВУ. Устройства для ввода	3 ч.

	информации: клавиатура, манипуляторы( мышь, джойстик, трекбол и т.д.), сканер, диджитайзеры, цифровые фотокамеры. Устройства для вывода информации: монитор, принтер, графопостроитель. Средства связи и телекоммуникации: модем, факс - модем. Запоминающие устройства ПК. Классификация и основные характеристики. Рекомендации по выбору ПК.	
4	Программное обеспечение ЭВМ. Программные продукты и их основные характеристики. Классификация программных продуктов. Сервисное программное обеспечение. Программы архиваторы, программы обслуживания магнитных дисков. Антивирусные программные средства.	2 ч.
5	Прикладные программные продукты. Текстовый процессор, Базовые возможности. Основные понятия. Копирования, перемещение, удаление текста. Форматирование текста. Работа с текстом. Табличный процессор EXCEL. Основные понятия: ссылка, абсолютная и относительная ссылка, адрес, ячейка, автозаполнение, диаграмма, лис, книга, формула, функция. Функциональные возможности. Технология работы в электронной таблице.	3 ч.
6	Базы данных. Основные понятия. Виды моделей данных. Иерархическая модель, сетевая модель, реляционная. Понятие информационного объекта. Типы связей: один -к- одному, один-ко-многим, многие - ко многим. Функциональные возможности БД. Запросы: мастер простой запрос, конструктор запросов, перекрестные запросы, универсальные запросы с параметром. Основы технологии работы в БД.	2 ч.
7	Алгоритмизация и программирование. Методика подготовки и решения задачи на ПК. Основные понятия теории алгоритмов. Способы записи алгоритма. Алгоритм и его свойства. Языки программирования высокого уровня.	2 ч.
8	Локальные и глобальные сети. Назначение и классификация компьютерных сетей. Характеристика процесса передачи данных. Архитектура компьютерных сетей. Протоколы компьютерной сети. Локальные вычислительные сети. Особенности организации ЛВС. Типовые топологии и методы доступа ЛВС. Глобальная сеть Internet. Представление о структуре и системе адресации. Способы организации передачи информации.	2 ч.
	<b>Итого</b>	18ч.

<b>Лабораторные работы</b>		
	<b>Специальность 040201 (36 ч.)</b>	
1	MS DOS. Работа с файлами и каталогами (создание, удаление, переименование, перемещение).	2 ч.
2	Контрольная работа	1 ч.
3	Total Commander. Структура окна. Верхнее меню. Выбор диска для панели. Основные виды панелей, их структура и назначение. Функциональные	2 ч.

	клавиши. Работа с файлами и каталогами (создание, удаление, переименование, перемещение). Управление при помощи верхнего меню.	
4	Контрольная работа	1 ч.
5	Операционная система Windows. Знакомство с элементами окна и способами его размеров. Конфигурирование оболочки Windows. Программа Поиск. Настройка главного меню. Программа Проводник. Работа с файлами, папками, ярлыками (создание, удаление, переименование, перемещение).	4 ч.
6	Контрольная работа	2 ч.
7	Текстовый редактор Word. Форматирование абзацев. Шрифт. Гарнитура и размер шрифта. Форматирование таблиц. Использование формул при вычислении в таблицах. Дополнительные возможности размещение графики в документе. Рисование в документе. Вставка рисунка из коллекции ClipArt. Текстовые эффекты. Редактор формул. Колонки. Шаблоны. Стили.	8 ч.
8	Контрольная работа	2 ч.
9	Табличный процессор Excel. Структура интерфейса. Панели инструментов. Основное меню. Ввод и редактирование данных. Формат данных. Ввод чисел и текста. Работа с функциями и формулами. Использование ссылок. Автоматическое изменение относительных ссылок при копировании и перемещении формул. Логические функции. Диаграммы и графики.	6 ч.
10	Введение в язык HTML. Устройство Web-страницы. Язык HTML. Описание документа с помощью языка HTML. Интерпретация тегов HTML. Основные структурные элементы HTML. Заголовки. Абзацы. Адресные данные. Отображение предварительно форматированного текста. Линейки. Списки: маркированные, нумерованные. Создание гиперссылок. Добавление графических элементов. Представление содержания документа в таблицах. Простая модель таблицы. Атрибуты тегов таблицы. Сложная модель таблицы. Управление представлением рамок и границ.	8 ч.
	<b>Итого</b>	<b>36ч.</b>

<b>Специальность 040101 (36 ч.)</b>		
1	MS DOS. Работа с файлами и каталогами (создание, удаление, переименование, перемещение).	2 ч.
2	Контрольная работа	1 ч.
3	Total Commander. Структура окна. Верхнее меню. Выбор диска для панели. Основные виды панелей, их структура и назначение. Функциональные клавиши. Работа с файлами и каталогами (создание, удаление, переименование, перемещение). Управление при помощи верхнего меню.	2 ч.
4	Контрольная работа	1 ч.
5	Операционная система Windows. Знакомство с элементами окна и способами его размеров. Программа Поиск. Настройка главного меню. Программа Проводник. Работа с файлами, папками, ярлыками (создание, удаление,	4 ч.

	переименование, перемещение). Архивация данных с помощью программы WinZip. Запуск. Внешний вид окна. Создание архивов. Извлечение файлов из архива.	
6	Контрольная работа	1 ч.
7	Текстовый редактор Word. Форматирование абзацев. Шрифт. Гарнитура и размер шрифта. Форматирование таблиц. Использование формул при вычислении в таблицах. Дополнительные возможности размещение графики в документе. Рисование в документе. Вставка рисунка из коллекции ClipArt. Текстовые эффекты. Редактор формул. Колонки. Шаблоны. Стили.	6 ч.
8	Контрольная работа	1 ч.
9	Табличный процессор Excel. Структура интерфейса. Панели инструментов. Основное меню. Ввод и редактирование данных. Формат данных. Ввод чисел и текста. Работа с функциями и формулами. Использование ссылок. Автоматическое изменение относительных ссылок при копировании и перемещении формул. Логические функции. Диаграммы и графики.	5 ч.
10	Контрольная работа	1 ч.
11	Работа с графическим редактором CorelDraw. Особенности работы в CorelDraw. Внешний вид окна. Настройка параметров CorelDraw. Совместное использование панелей инструментов, свойств, интерактивных средств. Работа с графикой и текстом.	4 ч.
12	СУБД ACCESS 7.0. Создание собственной БД. Мастер таблиц. Режим конструктора. Режим таблиц. Режим конструктора. Запросы. Мастер простой запрос. Конструктор запросов. Вычисление по полям. Формы	6 ч.
13	Контрольная работа	2 ч.
	<b>Итого</b>	<b>36ч.</b>

	<b>2 семестр. Специальность 040201 (54 ч)</b>	
1	СУБД ACCESS 7.0. Создание собственной БД. Мастер таблиц. Режим конструктора. Режим таблиц. Режим конструктора. Запросы. Мастер простой запрос. Конструктор запросов. Вычисление по полям. Формы.	4 ч.
2	Контрольная работа	1 ч.
3	Мастер презентаций PowerPoint. Создание и оформление презентаций. Создание и применение шаблонов презентаций. Эффектное оформление презентаций.	2 ч.
4	MathCAD. Внешний вид окна. Меню. Панели инструментов. Простейшие вычисления и операции. Арифметические вычисления. Определение переменной и ее значения. Вычисление значений выражений, содержащих переменные. Определение и вычисление значения функции в точке. Построение таблицы значений функции. Построение графиков функций.	1 ч.
5	Действия с матрицами. Основные матричные операции. Транспонирование. Вычисление обратной матрицы. Вычисление степени матрицы. Вычисление определителей. Решение системы линейных алгебраических уравнений.	1 ч.

6	Символьные вычисления с математическими выражениями. Вычисление пределов, производных, неопределенных и определенных интегралов. Решение алгебраических уравнений.	2 ч.
7	Контрольная работа	1 ч.
8	Работа с графическим редактором CorelDraw. Особенности работы в CorelDraw. Внешний вид окна. Настройка параметров CorelDraw. Совместное использование панелей инструментов, свойств, интерактивных средств. Работа с графикой и текстом.	3 ч.
9	Основные понятия языка: идентификатор, оператор, ключевое слово. Структура программных объектов (подпрограмм, модуля, программы). Стандартные типы данных языка. Арифметические операции, выражения и функции.	2 ч.
10	Знакомство с компонентами страницы Standard Delphi	4 ч.
11	Технология программирования задач линейной структуры	4 ч.
12	Программирование задач разветвляющейся структуры	4 ч.
13	Программирование задач с множественным выбором	5 ч.
14	Программирование задач циклической структуры	6 ч.
15	Программирование задач с данными типа вектор	6 ч.
16	Технология программирования задач с данными типа матрица	6 ч.
	<b>ВСЕГО</b>	<b>54ч.</b>

### **2.5. Темы для самостоятельного изучения**

1. История развития ПК.
2. Классификация ЭВМ.
3. Понятие файла, каталога, диска.
4. Windows элементы окна, стандартные обозначения.
5. Назначение программы Проводник.
6. Компьютер и здоровье.
7. Информация как объект права собственности.
8. Объекты защиты информации. Методы защиты информации.
9. Обзор современных методов защиты информации.
10. Ограничение доступа. Контроль доступа к аппаратуре

Контроль самостоятельной работы студентов осуществляется в письменной форме в виде опроса за 5 мин до окончания занятия.

### **3. Перечень промежуточных форм контроля знаний.**

Результативность работы обеспечивается системой контроля, которая при очной форме обучения включает опрос студентов на лабораторных занятиях, контрольные работы, выполнение лабораторных работ, проведение экзамена.

Студент допускается к экзамену при условии посещения всех лекционных и лабораторных занятий.

### 3.1. Вопросы к экзамену ( спец-ть 040101)

1. Информатика как наука. Этапы развития информатики.
2. Направления в информатике.
3. Информация, ее виды, свойства, измерение.
4. Системный блок, его внутреннее устройство.
5. Процесс обработки информации.
6. Устройства памяти. Типы памяти и их назначение.
7. Языки программирования.
8. Периферийные устройства ЭВМ их назначения.
9. Основные понятия теории алгоритмов, способы описания алгоритмов.
10. База данных. Их классификация.
11. Клавиатура, основные группы клавиш, их функции.
12. Прикладное программное обеспечение, основные направления.
13. Системное ПО.
14. Программное обеспечение. Классификация ПО.
15. Компьютерный вирус. Основные виды. Пути проникновения вирусов.
16. Форматирование символов, абзацев, страниц в Word.
17. Антивирусные программы. Меры по защите от вирусов.
18. Excel. Оформление таблицы. Форматирование столбцов и строк.
19. Компьютерные сети, основные понятия.
20. ЛВС, возможности, компоненты.
21. Топология ЛВС.
22. Иерархическая модель. Достоинства и недостатки.
23. Разновидности принтеров и их характеристика.
24. ГВС, возможности, компоненты.
25. Всемирная информационная компьютерная сеть.
26. Структуры алгоритмов (линейная, ветвление, циклическая, подпрограммы).
27. Основные этапы решения задач на ЭВМ.
28. Типы моделей организации баз данных. Их достоинства и недостатки.
29. Внешне запоминающие устройства. Классификация.
30. Операционная система Windows. Назначения, возможности, требования к аппаратуре при инсталляции.
31. Windows. Программа Проводник
32. Windows. Виды окон. Работа с окнами. Переход между окнами.
33. Excel. Данные, хранимые в ячейках ЭТ. Форматирование данных.

34. Excel. Ячейка. Ввод данных. Форматирование текста. Изменение ширины столбца и высоты Excel. Окно и его элементы.
35. Строки. Выделение.
36. Программа Поиск. Стандартные программы Windows.
37. Excel. Назначение. Основные понятия
38. Excel. Формулы и функции вводимые в ячейках ЭТ. Примеры.
39. Работа с файлами и каталогами. Выделение группы файлов.
40. Архивация файлов. WinZip.
41. Программная оболочка Norton Commander. Виды информации на панелях. Сортировка файлов.
42. Основные элементы графического интерфейса Windows - Рабочий стол, Панель задач, меню.
43. СУБД ACCESS. Создание таблиц в режимах: конструктора, таблиц, мастера.
44. Структурные элементы БД. Описание логической структуры записи файла
45. Запросы в СУБД ACCESS. Примеры.

### 3.2. Вопросы к зачету ( спец-ть 040201)

1. Информатика как наука. Этапы развития информатики.
2. Направления в информатике.
3. Информация, ее виды, свойства, измерение.
4. Системный блок, его внутреннее устройство.
5. Процесс обработки информации.
6. Устройства памяти. Типы памяти и их назначение.
7. Программная оболочка Total Commander . Основные возможности. Понятие файла, каталога, диска
8. Возможности Windows. Основные элементы, стандартные обозначения.
9. Основные элементы графического интерфейса Windows - Рабочий стол, Панель задач, меню.
10. Программа Проводник. Создание папки, файлов. Переименование, копирование, перенос, удаление файлов.
11. Windows. Виды окон. Работа с окнами. Переход между окнами.
12. Windows. Программа Проводник.
13. Промежуточный Буфер обмена. Программа Блокнот.
14. Панель управления. Задание собственной конфигурации.
15. Профессиональный текстовый редактор Word. Его возможности. Запуск. Внешний вид окна.
16. Использование таблиц в Word (вставка, форматирование, использование формул).
17. Клавиатура, основные группы клавиш, их функции.
18. Вставка и работа с объектами в Word.
19. Norton Commander. Работа с файлами и каталогами. Выделение группы файлов.
20. Введение в язык HTML. Устройство Web-страницы.
21. Язык HTML. Описание документа с помощью языка HTML.
22. Интерпретация тегов HTML. Символьные объекты. Элементы HTML и их вложение.
23. Основные структурные элементы HTML. Заголовки. Абзацы. Адресные данные. Логические разделы. Отображение предварительно форматированного текста.
24. Линейки. Списки. Маркированные списки. Нумерованные списки.
25. Создание гиперссылок. Добавление графических элементов.
26. Представление содержания документа в таблицах. Простая модель таблицы. Атрибуты тегов таблицы. Сложная модель таблицы. Управление представлением рамок и границ.

### 3.3. Вопросы к экзамену (спец-ть 040201)

1. Информатика—предмет и задачи. Появление и развитие информатики  
Структура информатики.
2. Информация и данные. Информация в материальном мире. Свойства информации
3. Передача информации. Общая характеристика процессов сбора, передачи обработки и накопления информации.
4. Представление информации в ЭВМ. Системы счисления.
5. Структурная и функциональная организация ЭВМ: микропроцессор, сопроцессор, память, контроллеры и шина, электронные платы, накопители.
6. Функциональные характеристики ПК.
7. Классификация внешних устройств ПК.
8. Устройства для ввода информации клавиатура, манипуляторы (мышь, джойстик, трекбол и т д), сканер, диджитайзеры, цифровые фотокамеры.
9. Устройства для вывода информации монитор, принтер, графопостроитель.
- 10.Классификация внешних и внутренних запоминающих устройств. Основные характеристики.
- 11.Классификация программного обеспечения ЭВМ. Программные продукты и их основные характеристики. Классификация программных продуктов.
- 12.Базовое программное обеспечение ПК. Понятия ОС- операционной системы, ОО – операционной оболочки. Базовые элементы ОС. Классификация ОС.
- 13.Сервисное программное обеспечение: программы архиваторы, программы обслуживания магнитных дисков.
- 14.Классификация вирусов и признаки заражения.
- 15.Антивирусные программы.
- 16.Классификация пакетов прикладных программ.
- 17.Прикладные программные продукты. Текстовый процессор. Команды Word:  
Параметры страницы, Форматирование абзаца, шрифта, списка, колонтитулов.  
Объекты (wordArt, ClipArt, Equation). Таблица. Проверка правописания.
- 18.Табличный процессор EXCEL. Виды адресации ячеек (относительная, абсолютная, смешанная).
- 19.Табличный процессор EXCEL. Форматирование ячеек (типы данных ячеек, границы, шрифт, заливка).

20. Табличный процессор EXCEL. Виды диаграмм.
21. Табличный процессор EXCEL. Функции СУММ, СРЗНАЧ, МАХ, МИН, ЕСЛИ, И, ИЛИ, НЕ.
22. Табличный процессор EXCEL. Анализ данных. Подбор параметра. Программа Поиск решения.
23. Табличный процессор EXCEL. Анализ данных. Программа Поиск решения.
24. База данных. Модели базы данных.
25. Основные элементы и объекты базы данных. Типы связей.
26. СУБД ACCESS. Создание таблиц в режимах: конструктора, таблиц, мастера.
27. Запросы в СУБД ACCESS. Примеры.
28. Особенности работы с CorelDraw.
29. Мастер презентаций PowerPoint. Режимы PowerPoint. Создание и оформление презентаций. Анимация. Управляющие кнопки. Переходы.
30. Назначение и классификация компьютерных сетей.
31. Компьютерные сети: аппаратные средства для передачи данных.
32. Архитектура компьютерных сетей.
33. Протоколы компьютерной сети: понятие протокола, типы протоколов.
34. Локальные вычислительные сети: устройства в сети (сервер, рабочая станция).
35. Локальные вычислительные сети: физическая передающая среда.
36. Локальные вычислительные сети: типовые топологии и методы доступа ЛВС.
37. Локальные вычислительные сети: способы объединения.
38. Глобальная сеть Internet: представление о структуре и системе адресации.
39. Службы Internet.
40. Глобальная сеть Internet: протоколы, программы-браузеры.
41. Защита сетевых ресурсов.
42. Средства автоматизации научно-исследовательских работ в системе MathCAD. Внешний вид окна MathCAD. Меню. Панели инструментов MathCAD.
43. Простейшие вычисления и операции в MathCAD. Арифметические вычисления. Определение переменной и ее значения. Вычисление значений выражений, содержащих переменные. Определение и вычисление значения функции в точке. Построение таблицы, значений функции.

44. Простейшие вычисления и операции в MathCAD. Действия с матрицами. Основные матричные операции. Транспонирование. Вычисление обратной матрицы. Вычисление степени матрицы. Вычисление определителей.
45. Простейшие вычисления и операции в MathCAD. Вычисление пределов, производных, неопределенных и определенных интегралов.
46. Алгоритм. Свойства алгоритма.
47. Моделирование решения задач. Этапы проектирования и создания программ.
48. Понятия язык и система программирования, 2 вида языков программирования. Компилятор, транслятор.
49. Поколения языков программирования.
50. Обзор языков программирования высокого уровня
51. Системы программирования
52. Основные понятия языка Object Pascal
53. Данные. Структура типов данных. Простые типы данных
54. Типы данных Object Pascal.
55. Математические выражения, функции и операции. Оператор присваивания
56. Составной оператор
57. Ветвление: понятие, операторы (2 вида), блок-схемы.
58. Операции сравнения ( $=$ ,  $>$ ,  $<$ ,  $<>$ ,  $<=$ ,  $>=$ ,  $=$ ). Логические связки (И, ИЛИ, НЕ).
59. Цикл: понятие, блок-схема цикла с пред- и постусловиями, оператор цикла с параметром.
60. Оператор выбора, блок-схема.
61. Зависимый и независимый переключатели (флажок и радиокнопки).
62. Массивы. Описание вектора в Object Pascal, обращение к элементами вектора.
63. Массивы. Двумерные массивы
64. Компоненты TcheckBox, TradioGroup, Tpanel, TEdit, TLabel, Tmemo, TButton, TRadiogroup, TlistBox.

#### 4. Учебно-методические материалы

##### 4.1. Основная литература

1. **Информатика. Базовый курс** [Текст]: учеб.: рек. Мин. обр. РФ / Ред. С.В. Симонович. - СПб.: Питер, 2000, 2004, 2005, 2006. - 638 с.
2. **Информатика** [Текст]: учеб.: рек. Мин. обр. РФ/ под ред. Н. В. Макаровой. - М.: Финансы и статистика, 2000, 2005, 2001. - 268 с.
3. **Информатика** [Текст]: практикум по технологии работы на компьютере: рек. Мин. обр. РФ / под ред. Н. В. Макаровой. - М.: Финансы и статистика, 2005. - 256 с.
4. **Могилев, А. В.** Информатика [Текст]: УЧЛ - Учебное пособие / А. В. Могилев, Н. И. Пак, Е. К. Хеннер / под ред. Е. К. Хеннер. - М.: Академия, 2004. - 842 с.
5. **Бобровский, С. И.** Delphi 7 [Текст]: Учебный курс / С. И. Бобровский, 2004, 2006. - 736 с.

##### 4.2. Дополнительная литература

1. **Гиляревский, Р. С.** Основы информатики [Текст]: Курс лекций / Р. С. Гиляревский. - М.: Экзамен, 2003. - 320 с.
2. **Информатика для юристов и экономистов** [Текст]: Учебник / Ред. С.В. Симонович. - СПб. : Питер, 2001, 2003. - 688 с.
3. **Ляхович, В. Ф.** Основы информатики [Текст]: учеб. пособие: Рек. Мин. обр. РФ / В.Ф. Ляхович, С.О. Крамаров. - 3-е изд. - Ростов н/Д : Феникс, 2003. - 700 с.
4. **Могилев, А. В.** Практикум по информатике [Текст]: учеб. пособие / А.В. Могилев, Н.И. Пак, Е.К. Хеннер; Под ред. Е.К. Хеннера. - М. : Академия, 2002. - 608 с.
5. **Острейковский, В. А.** Информатика [Текст]: Учеб.: Рек. Мин. обр. РФ / В.А. Острейковский. - М. : Высш. шк., 1999. - 512 с.
6. **Фаронов, В. В.** Система программирования Delphi [Текст]: производственно практическое издание / В.В. Фаронов, 2003.
7. **Культин, Н.** Delphi 3. Программирование на Object Pascal [Текст]: производственно-практическое издание / Н. Культин, 1998. - 304 с.
8. **Макарчук, Т.А.** Основы программирования в Delphi 5 [Текст]: Учеб.-метод. пособие / АмГУ, ФМиИ, 2002. - 110 с.
9. **Макарчук, Т.А.** Основы программирования в Delphi [Текст]: лаб. практикум / АмГУ, ФМиИ, 2005. - 81 с. <http://libserv.amursu.ru/DiBooks/e-copy/400.pdf>.

## **II. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРОФЕССОРСКО-ПРЕПОДАВАТЕЛЬСКОМУ СОСТАВУ**

### **1. Методические рекомендации по проведению лекционных занятий.**

Задача лекции состоит не столько в изложении системы теоретических знаний, сколько в общении с аудиторией, сообщении ей смысла и значения излагаемого материала, в дальнейшем развитии знания. Полученные в ходе лекции знания, часто носят характер поверхностного усвоения, при этом должны служить дальнейшим мотивом и основой для дальнейшей организации самостоятельной учебно-познавательской деятельности по приобретению новых знаний и умений, приводящей к глубокому освоению понятий, как отдельной темы, так и науки в целом.

На очном лекционном занятии в вводной части определяются минимальные знания, умения и навыки, подлежащие усвоению в ходе изучения темы курса. В основной части рассматривается довольно большой объем материала, в основном, обзорного характера. В заключительной части лекции излагается постановка типовых задач темы, решение которых подробно будет рассмотрено на лабораторных занятиях.

Изложение информационного материала лекции предполагает использование объяснительно-иллюстративного метода с применением фронтальной формы организации обучения.

### **2. Методические рекомендации по проведению лабораторных работ.**

На лабораторных занятиях по информатике формируется и совершенствуется практический уровень владения информационными процессами, основывающийся на применении теоретических знаний. Для проведения лабораторных занятий со студентами по дисциплине «Информатика» используются компьютерные классы. Занятия в компьютерном классе предполагают индивидуальную или парно-групповую формы организации обучения.

Этапы проведения лабораторной работы следующие:

- Контрольный опрос студентов для проверки готовности к выполнению лабораторной работы (до 10 мин).
- Выдача индивидуального задания и пояснения о порядке выполнения индивидуального задания (до 5 мин).
- Выполнение индивидуального задания (около 1 ч.)
- Оформление результатов работы. Сдача выполненной работы преподавателю (до 10 мин).
- Получение домашнего задания (1-2 мин.)

- Приведение в порядок рабочего места, в том числе закрытие всех рабочих окон и уничтожение созданных на винчестере индивидуальных файлов (3-4 мин).

Индивидуальные задания для лабораторных работ должны быть представлены конкретно-практическими и творческими задачами.

На первой ступени изучения темы выполняются конкретно-практические задачи, при решении которых формируется минимальный набор умений. Преподаватель опосредованно руководит самостоятельной познавательной деятельностью студентов, консультирует студентов при возникновении непосильных затруднений в ходе решения задачи, обращает внимание группы "опасные" места решения. Отработка минимального набора навыков завершается во внеаудиторное время при выполнении домашней работы. Принимая во внимание сложность доступа некоторыми студентами к компьютерной технике во внеаудиторное время, домашние задания по "Информатике" должны носить большей части моделирующий характер.

Вторая ступень изучения темы дифференцируется в зависимости от степени усвоения его обязательного уровня. Студенты, овладев основами теории и усвоив содержание типовых методов и приемов решения задач, приступают к решению творческих задач. Если уровень знаний и умений, демонстрируемых студентом при контрольном обследовании, не соответствует установленным требованиям, студент вновь возвращается к стандартным упражнениям, но под более пристальным наблюдением преподавателя.

По завершению изучения отдельной темы курса по результатам выполнения лабораторных работ каждый студент получает оценку.

Студенты, пропустившие лабораторные занятия, должны их выполнить во внеаудиторное время и отчитаться до начала зачетно-экзаменационной сессии.

### **3. Методические рекомендации по организации контроля знаний студентов**

В Университете качество освоения образовательных программ оценивается путем осуществления текущего контроля успеваемости, проведения промежуточных аттестаций и итогового контроля по окончании семестра.

На первом занятии до сведения студентов доводятся требования и критерии оценки знаний по дисциплине.

Целью текущего контроля успеваемости является оценка качества освоения студентами образовательных программ в течение всего периода обучения. К главной задаче текущего контроля относится повышение мотивации студентов к регулярной учебной работе, самостоятельной работе, углублению знаний, дифференциации итоговой оценки знаний.

Текущий контроль успеваемости осуществляется систематически и, как правило, преподавателем, ведущим лабораторные занятия. Формами текущего контроля являются

письменные опросы, автоматизированное тестирование, аудиторные контрольные работы, отчеты по лабораторным работам, домашние задания. В течение семестра преподавателем должно быть проведено не менее 7-ми контрольных проверок знаний по каждому студенту из учебной группы.

Результаты текущего контроля служат основанием для прохождения студентом промежуточной аттестации.

Итоговый контроль (зачет или экзамен) по информатике преследуют цель оценить работу студентов за курс, полученные теоретические знания, их прочность, развитие творческого мышления, навыки самостоятельной работы, умение синтезировать полученные знания и применять их при решении практических задач. Задания итогового контроля состоят из двух частей: письменного теоретического опроса (от 6 до 12 вопросов) и практических заданий (от 1 до 3), выполняемых на компьютере.

Во время проведения итогового контроля (зачета или экзамена) студентам не разрешается пользоваться вспомогательными материалами Их использование, а также попытки общения с другими студентами или иными лицами, в т.ч. с применением электронных средств связи, перемещения без разрешения экзаменатора и т.д., являются основанием для удаления студента из аудитории с последующим выставлением в ведомость неудовлетворительной оценки.

Итоговая оценка по дисциплине рассчитывается по формуле:  $0,4x+0,6y$ , где  $x$  - средняя оценка, полученная в результате выполнения текущих форм контроля,  $y$  - результат итогового зачета (экзамена).

## КОНСПЕКТ ЛЕКЦИЙ

### ЛЕКЦИЯ № 1

**Тема: Информатика. Информация.**

**Информатика** – это наука, изучающая все аспекты получения, хранения, переработки, передачи и использования информации.

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

*Главная функция* информатики заключается в разработке методов и средств преобразования информации и их использовании в организации технологического процесса переработки информации.

*Основная задача* информатики заключается в определении общих закономерностей, в соответствии, с которыми происходит создание научной информации, ее преобразование,

передача и использование в различных сферах деятельности человека. Прикладные задачи заключаются в разработке более эффективных методов и средств осуществления информационных процессов, в определении способов оптимальной научной коммуникации с широким применением технических средств.

Таким образом, **задачами информатики** являются:

1. Исследование информационных процессов любой природы.
2. Разработка информационной техники и создание новейшей технологии переработки информации на базе полученных результатов исследования информационных процессов.
3. Решение научных и инженерных проблем создания, внедрения и обеспечения эффективного использования компьютерной техники и технологии во всех сферах общественной жизни.

### **Основные направления информатики:**

1. *Теоретическая информатика* – математическая дисциплина, которая использует методы математики для построения и изучения моделей обработки, передачи и использования информации.
2. *Кибернетика* – это наука об общих принципах управления в различных системах: технических, биологических, социальных и др.
3. *Программирование* – создание новых языков программирования.
4. *Искусственный интеллект* – это направление определяет стратегическое направление развития информатики. Основная цель работы в этой области – стремление проникнуть в тайны творческой деятельности людей и их способности к овладению науками, знаниями и умениями.
5. *Информационные системы* – это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.
6. *Вычислительная техника* – представляет собой самостоятельное направление исследований.
7. *Информатика в обществе* - информационная среда, которая обеспечивает доступ ко всей необходимой информации, накопленной человеком.
8. *Информатика в природе* – изучение информационных процессов, протекающих в биологических системах, и использование накопленных знаний при организации и управлении природными системами и создании технических средств.

### **Этапы развития информатики:**

1. *Ручной* – с 50-х годов до н. э.

2. *Механический* – с середины 17 века
3. *Электромеханический*. Первый счетно-аналитический комплекс был создан в США в 1887 г.
4. *Электронный* – с 40-х гг. 20 века.

### Классификация ЭВМ:

#### 1. По истории создания:

I поколение: элементарная база — лампы, оперативная память на электронно-лучевых трубках и ферритовых сердечниках, быстродействие до 20000 оп/сек., охлаждение, однопрограммность.

II поколение: элементарная база — полупроводниковые транзисторы, быстродействие  $10^4$ – $10^5$  оп/сек. Объем памяти — до 150 слов при длине слова до 50 двоичных разрядов. Программирование велось на алгоритмических языках Фортран, Алгол, Кобол.

III поколение: элементарная база — интегральные схемы (ИС), быстродействие  $10^6$ – $10^7$  оп/сек. Резко снижены габариты и энергопотребление ЭВМ. Оперативная память строилась на ИС и достигала объема  $10^5$ – $10^6$  байт. Унифицировались периферийные устройства. Появился широкий выбор языков программирования. Стали использоваться операционные системы, позволяющие резко повысить производительность и организовать многопрограммный и терминальный режимы.

IV поколение: элементарная база — большие и сверхбольшие ИС (БИС и СБИС). Быстродействие  $10^7$ – $10^8$  оп/сек. Формируются два направления — многопроцессорные и персональные ЭВМ. Появляются компьютерные сети. Разрабатывается специализированное программное обеспечение, позволяющее оперативно программировать решение задач определенного класса (например, в таких областях как статистика, инженерная графика, научно-технические расчеты и т.д.).

#### 2. По размерам: суперЭВМ, большие ЭВМ, малые ЭВМ, микроЭВМ

Параметры	СуперЭВМ	Большие ЭВМ	Малые ЭВМ	МикроЭВМ
Производительность, MIPS	1000-100000	10-1000	1- 100	1- 100
Емкость ОЗУ	2000-10000	64-10000	4-512	4-256
Емкость винчестера, Гбайт	500 – 5000	50 - 1000	2- 100	0,5 - 10
Разрядность, бит	64-128	32-64	16-64	16-64

*СуперЭВМ* – самые мощные вычислительные системы, интенсивно развивающиеся в настоящее время. Используются для решения ряда задач: прогнозирования метеобстановки, управление сложными оборонными комплексами, моделирования экологических систем и др.

*Большие ЭВМ* отличаются несколько меньшими характеристиками. Первоначально появились большие ЭВМ в 1946 г.

Появление в 70-х гг. *малых ЭВМ* связано прогрессом в области ВТ и избыточностью ресурсов больших ЭВМ для ряда приложений. Они более компактны и значительно дешевле больших ЭВМ.

*Микро ЭВМ* - вычислительные машины, меньшие по размеру и стоимости малых ЭВМ, но по производительности сравнимые с малыми ЭВМ. К классу микроЭВМ относятся персональные компьютеры.

### **Развитие отечественной вычислительной техники**

В бывшем СССР работы по созданию ЭВМ были начаты перед Великой Отечественной войной. Однако работы в этом направлении из-за войны были приостановлены.

Разработка ЭВМ возобновилась в 1947 г. в Институте электротехники Академии наук Украины под руководством Сергея Алексеевича Лебедева.

В декабре 1948 г. С.А. Лебедевым (независимо от Джона фон Неймана) были разработаны принципы построения ЭВМ, у которой программа хранилась в оперативной памяти. К концу 1949 г. были спроектированы общая компоновка машины и принципиальные схемы ее блоков. В первой половине 1950 г. были изготовлены отдельные блоки и к концу 1950 г. закончена отладка созданного макета. В ноябре 1950 г. был испытан макет первой отечественной ЭВМ — малой электронно-счетной машины (МЭСМ). В 1952 г. она была введена в эксплуатацию.

В 1952 г. была создана большая электронно-счетная машина (БЭСМ). В качестве элементной базы у этой машины использовались электронные лампы (первое поколение ЭВМ).

Первая отечественная ЭВМ на полупроводниковых приборах (второе поколение ЭВМ) под названием «Днепр» была разработана в конце 50-х годов в Институте кибернетики АН Украины под руководством академика В.М. Глушкова.

ЭВМ второго поколения БЭСМ-6 (1966), разработанная под руководством С.А. Лебедева, была одной из самых производительных машин в мире.

В бывшем СССР первым серийным компьютером на интегральных микросхемах (третье поколение ЭВМ) была машина «Наири-3», появившаяся в 1970 г.

К четвертому поколению можно отнести отечественные ЭВМ: ЕС-1015, ЕС-1025, ЕС-1035, ЕС-1045, ЕС-1055, ЕС-1065 («Ряд 2»), ЕС-1036, ЕС-1046, ЕС-1066, СМ-1420, СМ-1600, СМ-1700, персональные ЭВМ («Электроника МС 0501», «Электроника-85», «Искра-226»),

ЕС-1840, ЕС-1841, ЕС-1842, «Нейрон И9.66» и др.), многопроцессорный вычислительный комплекс «Эльбрус» и другие ЭВМ.

В конце 80-х — начале 90-х годов XX столетия в России были популярны бытовые персональные компьютеры «Микроша», «Радио-86», «Мик-ро-88», «Криста», «Лик», «Специалист», «Квант» (на процессоре K580BM80A), БК-0010 (на процессоре K1801BM1), «Ассистент», «Поиск», МС-0511, МС-1502 (на процессоре K1810BM86).

В середине 2001 года в России введён в строй 768-процессорный суперкомпьютер МВС-ЮООМ, обеспечивающий производительность в 1 Те-рафлопс. После этого Россия вышла на третье место в мире по мощности производимых суперкомпьютеров.

### **Классификация РС:**

1. *Стационарные.*

2. *Переносные:*

- а) *портативные* – выпускаются в виде дипломата (5-10 кг), имеют устройства: CD ROM, мультимедиа;
- б) *блокноты* – схожи с портативными РС, меньшего размера и памяти;
- в) *электронные словари* – электронные справочники для хранения имен, адресов, номеров телефонов, распорядка дня работы и т. д.
- д) *органайзеры (электронные записные книжки)* – обладают вместительной памятью, функциями работы с текстом и его редактирования.

### **Информационное общество, информационная культура, информационные системы и информационные технологии**

**Информационное общество** – это общество, в котором большинство работающих занято производством, хранением, переработкой или реализацией информации, особенно высшей ее формой – знаний.

Информационное общество характеризуется – 50% людей занято в сфере информационных услуг.

**Информационная культура** - умение целенаправленно работать и использовать информацию для получения, обработки и передачи с помощью современных технологий.

**Информационная система** – взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

**Информационная технология** – процесс, использующий совокупность средств и методов отбора, обработки и передачи данных для получения информации нового качества.

### **Понятия «сигнал» и «данные»**

Сигнал - физический процесс, имеющий информационное значение, установленное принятым соглашением. В обычной жизни под **сигналом** понимают физический процесс, который человек воспринимает как звук, свет и т. д.

**данные** – зарегистрированные сигналы. Данные могут рассматриваться как признаки или записанные наблюдения, которые по каким-то причинам не используются, только хранятся. Если появляется возможность использовать эти данные для уменьшения неопределенности в чем-либо, то данные превращаются в *информацию*.

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

Одной из важнейших разновидностей информации является информация экономическая.

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

#### **Свойства информации:**

1. *Объективность.*
2. *Достаточность (полнота)* – характеризует качество информации и достаточность данных для принятия решений или для создания новых данных.
3. *Достоверность* – если полезный сигнал зарегистрирован более четко, чем остальные сигналы, то достоверность информации более высока.
4. *Доступность* – мера возможности получить ту или иную информацию. Отсутствие доступа к информации приводит к одинаковому результату – информация не доступна.
5. *Актуальность* – степень соответствия информации текущему моменту времени.
6. *Адекватность* – степень соответствия информации, полученной потребителем тому, что автор вложил в ее содержание.
7. *Своевременность.*
8. *Полнота* – информации достаточно для принятия решения.
9. *Энтропия информации* – характеризует данные, среднее количество информации.

## Классификация информации по разным признакам:



## Передача и обработка информации

Различают две формы представления информации — **непрерывную (аналоговую) и прерывистую (цифровую, дискретную)**.

*Непрерывная форма* характеризует процесс, который не имеет перерывов и теоретически может изменяться в любой момент времени и на любую величину. Цифровой сигнал может изменяться лишь в определенные моменты времени и принимать лишь заранее обусловленные значения. Моменты возможного изменения уровня цифрового сигнала задает тактовый генератор конкретного цифрового устройства.

Для преобразования аналогового сигнала в цифровой сигнал требуется провести дискретизацию непрерывного сигнала во времени, квантование по уровню, а затем кодирование отобранных значений.

*Дискретизация* – замена непрерывного (аналогового) сигнала последовательностью отдельных во времени отсчетов этого сигнала.

### Количество информации

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

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

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

За единицу информации принимается один бит. Это такое количество информации, получается в результате реализации одного из двух равновероятных событий, например, при бросании монеты.

Применительно к ЭВМ на одном проводе можно реализовать два взаимоисключающих события: есть напряжение и нет напряжения. Следовательно, одним проводом можно передать 1 бит информации.

Более крупная единица информации — *байт* — равна 8 бит. Еще более крупная единица информации — 1 Кбайт — равна 1024 байтам. Далее— 1 Мбайт равен 1024 Кбайтам, 1 Гбайт равен 1024 Мбайтам, а 1 Тбайт равен 1024 Гбайтам.

Для измерения больших объемов информации применяют кратные единицы информации:

1 байт = 8 бит;

1 килобайт (Кб) = 1024 байт;

1 Мегабайт (Мб) = 1024 Кб = 1048576 байт;

1 Гигабайт (Гб) = 1024 Мб = 1048576 Кб.

Задача 1. Решить систему уравнений (найти  $x$ ,  $y$ ). В ответе необходимо указать единицы измерения.

$$\begin{cases} 5y - 2x = 7 \text{ Кбайт} \\ 4x = 2^{14} \text{ байт} \end{cases}$$

Решение.

$$5y - 2x = 7 \text{ Кбайт}$$

$$4x = 2^{14} \text{ байт}$$

$\Rightarrow$

$\cdot$

$$5y - 2x = 7 \text{ Кбайт}$$

$$4x = 2^{10} \cdot 2^4 \text{ байт}$$

$\Rightarrow$

$\cdot$

$$5y - 2x = 7 \text{ Кбайт}$$

$$4x = 2^4 \text{ Кбайт}$$

$\Rightarrow$

$\cdot$

$$5y - 2x = 7 \text{ Кбайт}$$

$$4x = 16 \text{ Кбайт}$$

$\Rightarrow$

$\cdot$

$$y = 3 \text{ Кбайт}$$

$$x = 4 \text{ Кбайт}$$

$\cdot$

$$\{ \cdot \cdot \cdot$$

$\cdot$

Задача 2. Одна страница книги содержит 25 строк по 80 символов. В книге 300 страниц. Каков объем информации необходим для хранения книги?

Задача 3. Один музыкальный файл занимает 6 Кб на диске. Сколько файлов можно записать на CD-диск объемом 600 Кб.

## ЛЕКЦИЯ № 2

**Тема: Аппаратное обеспечение ПК**

**Понятие персонального компьютера его архитектуры и структуры**

Современный компьютер состоит из двух взаимосвязанных частей – аппаратного (Hardware) и программного (Software) обеспечения. К аппаратному обеспечению относится все то, что можно потрогать руками (микросхемы, диски, платы).

*Архитектура компьютера* определяется совокупностью ее свойств, существенных для пользователя. Основное внимание при этом уделяется структуре и функциональным возможностям машины, которые можно разделить на основные и дополнительные.

**Персональный компьютер** — это настольная или переносная ЭВМ, удовлетворяющая требованиям общедоступности и универсальности применения.

## **Базовая аппаратная конфигурация ЭВМ**

### **Основные блоки и их назначения**

Существует понятие *базовой конфигурации*, которую считают типовой. Понятие базовой конфигурации может меняться. В настоящее время в базовой конфигурации рассматривают четыре устройства: системный блок, монитор, клавиатуру, мышь.

**Системный блок** представляет собой основной узел, внутри которого установлены наиболее важные компоненты. Устройства, находящиеся внутри системного блока, называют *внутренними*, а устройства, подключаемые к нему снаружи, называют *внешними*. Внешние дополнительные устройства, предназначенные для ввода, вывода и длительного хранения данных, также называют *периферийными*.

Корпуса ПК поставляются вместе с блоком питания, таким образом, мощность блока питания также является одним из параметров корпуса.

*Системный блок* обычно включает в себя системную плату, блок питания, накопители на дисках, разъемы для дополнительных устройств и платы расширения с контроллерами – адаптеры внешних устройств.

На системной плате (часто ее называют *материнской платой* (Mother Board)) размещаются самые главные детали:

1. *Микропроцессор (МП)* - это центральный блок ПК, предназначенный для управления работой всех блоков машины. В его состав входят:

- *устройство управления (УУ)* — формирует и подает во все блоки машины в нужные моменты времени определенные сигналы управления (управляющие импульсы); формирует адреса ячеек памяти, используемых выполняемой операцией, и передает эти адреса в соответствующие блоки ЭВМ; опорную последовательность импульсов устройство управления получает от генератора тактовых импульсов. В УУ входят *регистр команд* – запоминающий регистр, в котором хранится код команды, *дешифратор операций* и т.д.

- *арифметико-логическое устройство (АЛУ)* - предназначено для выполнения всех арифметических и логических операций над числовой и символьной информацией.
- *математический сопроцессор* - используется для ускоренного выполнения операций числами, для вычисления некоторых трансцендентных, в том числе тригонометрических, функций. Математический сопроцессор имеет свою систему команд и работает параллельно (совмещение во времени) с основным МП, но под управлением последнего.
- *микропроцессорная память (МПП)* - служит для кратковременного хранения, записи и выдачи информации, непосредственно используемой в вычислениях в ближайшие такты работы машины.
- *регистры* - быстродействующие ячейки памяти различной длины;
- *интерфейсная система микропроцессора* - реализует сопряжение и связь с другими устройствами ПК;

2. *Генератор тактовых импульсов* - он генерирует последовательность электрических импульсов; частота генерируемых импульсов определяет тактовую частоту машины.

*Частота генератора тактовых импульсов является одной из основных характеристик ПК и во многом определяет скорость его работы, ибо каждая операция в машине выполняется за определенное количество тактов.*

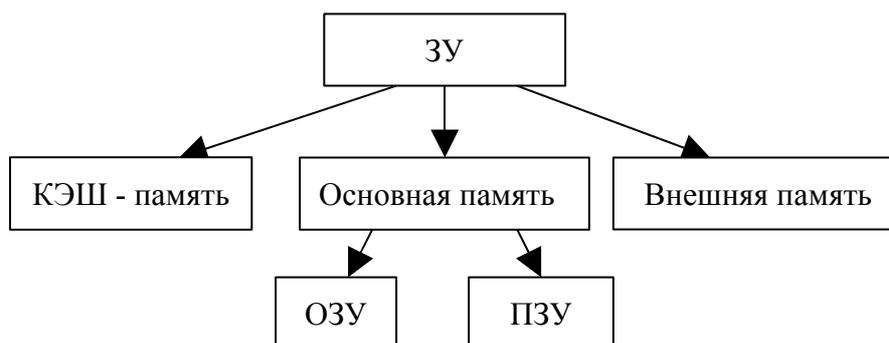
3. *Источник питания* - это блок, содержащий системы автономного и сетевого энергопитания ПК.

4. *Таймер* - это внутримашинные электронные часы, обеспечивающие при необходимости автоматический съём текущего момента времени (год, месяц, часы, минуты, секунды и доли секунд).

5. *Системная шина* - это основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой. Системная шина обеспечивает три направления передачи информации:

- 1) между микропроцессором и основной памятью;
- 2) между микропроцессором и портами ввода-вывода внешних устройств;
- 3) между основной памятью и портами ввода-вывода внешних устройств.

## Запоминающие устройства ПК



*Регистровая КЭШ-память* — высокоскоростная память сравнительно большой емкости, являющаяся буфером между ОП и МП и позволяющая увеличить скорость выполнения операций. В КЭШ-памяти хранятся данные, которые МП получил и будет использовать в ближайшие такты своей работы. Быстрый доступ к этим данным и позволяет сократить время выполнения очередных команд программы.

По принципу записи результатов различают два типа КЭШ-памяти: *КЭШ-память "с обратной записью"* и *КЭШ-память "со сквозной записью"*.

*Основная память* делится на оперативное (RAM) и постоянное (ROM) запоминающие устройство.

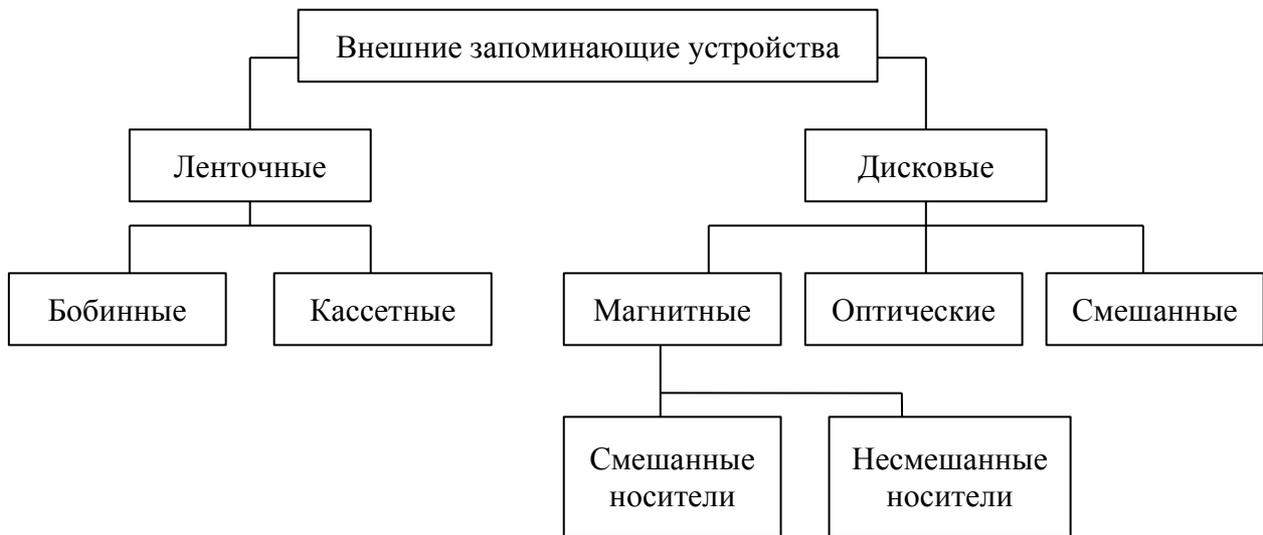
*Оперативное запоминающее устройство* предназначено для хранения информации (программ и данных), непосредственно участвующей в вычислительном процессе на текущем этапе функционирования ПК. ОЗУ — *энергозависимая память*: при отключении напряжения питания информация, хранящаяся в ней, теряется.

*Постоянное запоминающее устройство* также строится на основе установленных на материнской плате модулей (кассет) и используется для хранения неизменяемой информации: загрузочных программ операционной системы, программ тестирования устройств компьютера и некоторых драйверов базовой системы ввода-вывода (BIOS — Base Input-Output System) и др. Из ПЗУ можно только считывать информацию. ПЗУ — *энергонезависимое* запоминающее устройство.

Структурно основная память состоит из миллионов отдельных ячеек памяти емкостью 1 байт каждая. Общая емкость основной памяти современных ПК обычно лежит в пределах от 1 до 32 Мбайт. Емкость ОЗУ на один-два порядка превышает емкость ПЗУ: ПЗУ занимает 128 (реже 256) Кбайт, остальной объем — это ОЗУ.

Устройства *внешней памяти* или *внешние запоминающие устройства* (ВЗУ) разнообразны.

### Классификация внешних запоминающих устройств:



Диски относятся к машинным носителям с прямым доступом, это означает, что ПК может «обратиться» к дорожке, начинается участок с искомой информацией или куда нужно записать нужную информацию, где бы ни находилась головка записи/чтения накопителя.

Накопители на дисках очень разнообразны.

*Магнитные диски* (МД) относятся к магнитным машинным носителям информации. В качестве запоминающей среды у них используются магнитные материалы со специальными свойствами, позволяющими фиксировать два магнитных состояния. Каждому из этих состояний ставятся в соответствие двоичные цифры: 0 и 1. Диски бывают жесткими и гибкими, сменными и встроенными в ПК. Устройство для чтения и записи информации на магнитном диске называется *дисководом*.

Все диски характеризуются своим диаметром или, иначе, *формфактором*.

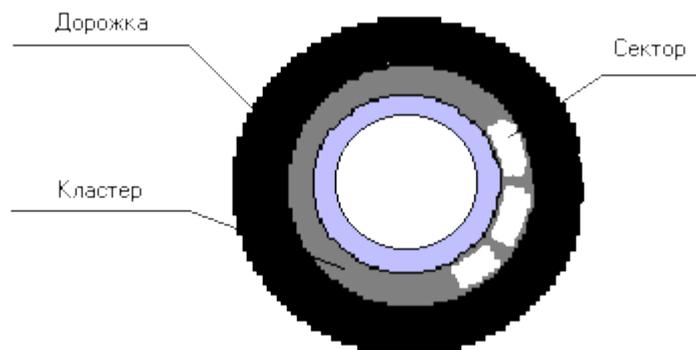


Рис.4.10 Логическая структура поверхности магнитного диска.

Информация на МД записывается и считывается *магнитными головками* вдоль концентрических окружностей — *дорожек*.

Каждая дорожка МД разбита на *сектора*. В одном секторе дорожки обычно помещено 512 байт данных. Обмен данными между НМД и ОП осуществляется последовательно целым числом секторов. *Кластер* — это минимальная единица размещения информации на диске, состоящая из одного или нескольких смежных секторов дорожки.

При записи и чтении информации МД вращается вокруг своей оси, а механизм управления магнитной головкой подводит ее к дорожке, выбранной для записи или чтения информации.

Данные на дисках хранятся в *файлах*, которые обычно отождествляют с участком памяти на этих носителях информации.

### **Накопители на гибких магнитных дисках**

На гибком магнитном диске (дискете) емкостью 1,44 Мб магнитный слой наносится на гибкую основу. МД помещают в пластмассовую кассету для защиты от пыли и механических повреждений.

Дискета изготавливается из гибкого пластика (лавсана), покрытого износостойчивым ферролаком, и помещается в футляр-конверт. Дискета имеет две прорези: центральное отверстие для соединения с дисководом и смещенное от центра небольшое отверстие (обычно скрытое футляром), определяющее радиус-вектор начала всех дорожек на ГМД. Футляр также имеет несколько прорезей: центральное отверстие, чуть большее, чем отверстие на дискете; широкое окно для считывающих и записывающих магнитных головок и боковую прорезь в виде прямоугольника, закрытие которой липкой лентой, например, защищает дискету от записи и стирания информации.

Каждую новую дискету в начале работы с ней следует отформатировать.

**Форматирование** дискеты — это создание структуры записи информации на ее поверхности: разметка дорожек, секторов, записи маркеров и другой служебной информации.

### **Накопители на жестких магнитных дисках**

В качестве накопителей на жестких магнитных дисках широкое распространение в ПК получили накопители типа "винчестер".

В этих накопителях один или несколько жестких дисков, изготовленных из сплавов алюминия или из керамики и покрытых ферролаком, вместе с блоком магнитных головок считывания/записи помещены в герметически закрытый корпус. Емкость этих накопителей благодаря чрезвычайно плотной записи, получаемой в таких несъемных конструкциях, достигает нескольких тысяч мегабайт.

Винчестер содержит диск или несколько дисков, расположенных в виде стопки один над другим, как блины.

Причина, по которой жесткий диск является "жестким", заключается в том, что он вращается с очень большой скоростью - обычно 3600 об/мин. Именно благодаря высокой скорости вращения диска считывание и запись информации осуществляются так быстро.

Жесткие диски стационарны. В настоящее время созданы съемные жесткие диски. Но обычно диски фиксируются внутри компьютеров (почему их иногда называют несъемными или стационарными дисками).

### **Накопители на оптических дисках**

В последние годы все большее распространение получают накопители на оптических дисках (НОД). Благодаря маленьким размерам, большой емкости и надежности эти накопители становятся все более популярными.

*Неперезаписываемые лазерно-оптические диски* обычно называют компакт-дисками ПЗУ —CD-ROM. Эти диски поставляются фирмой-изготовителем с уже записанной на них информацией. Запись информации на них возможна только вне ПК, в лабораторных условиях, лазерным лучом большой мощности, который оставляет на активном слое след — дорожку с микроскопическими впадинами. В оптическом дисководе ПК эта дорожка читается лазерным лучом существенно меньшей мощности.

*Перезаписываемые лазерно-оптические диски* с однократной и многократной записью. На этих дисках—CD-RW лазерный луч непосредственно в дисководе компьютера при записи прожигает микроскопические углубления на поверхности диска под защитным слоем; чтение записи выполняется лазерным лучом так же, как и у CD-ROM.

## Внешние устройства

Внешние устройства ПК обеспечивают взаимодействие машины с окружающей средой: пользователями, объектами управления и другими ЭВМ. ВУ весьма разнообразны и могут быть классифицированы по ряду признаков. Так, по назначению можно выделить следующие виды ВУ:

- устройства хранения данных (ВЗУ или внешняя память ПК);
- устройства ввода информации;
- устройства вывода информации;
- средства связи и телекоммуникации;
- устройства речевого ввода-вывода.

**1. Устройства ввода информации** - устройства, предназначенные для ввода информации в ПК.

1) Клавиатура — устройство для ручного ввода числовой, текстовой и управляющей информации в ПК. С помощью клавиатуры управляют компьютерной системой.

2) Графические планшеты (диджитайзеры) - для ручного ввода графической информации, изображений путем перемещения по планшету специального указателя (пера).

3) Сканеры (читающие автоматы) — для автоматического считывания с бумажных носителей и ввода в ПК машинописных текстов, графиков, рисунков, чертежей.

4) Цифровые фотокамеры. Как и сканеры, эти устройства воспринимают графические данные с помощью приборов с зарядовой связью, объединенных в прямоугольную матрицу.

5) Манипуляторы (устройства указания) Они предназначены для ввода графической информации на экран дисплея путем управления движением курсора по экрану с последующим кодированием координат курсора и вводом их в ПК:

- *джойстик*—рычаг;
- *мышь* представляет собой коробочку, снабженную двумя или тремя кнопками и легко уместящуюся в ладони. С помощью провода она подключается к порту в задней части системного блока
- *трекбол*— шар в оправе, в отличие от мыши он устанавливается стационарно, и его шарик приводится в движение ладонью руки.
- *световое перо*;
- *инфракрасная мышь* отличается от обычной наличием устройства беспроводной связи с системным блоком и т.д.

**2. Устройства вывода информации.**

**Принтеры** — печатающие устройства для регистрации информации на бумажный носитель.

1) Матричные принтеры.

*Принцип печати* этих принтеров состоит в том, что печатающая головка принтера, содержащая вертикальный ряд тонких металлических стержней (иглонок), движется вдоль печатаемой строки, а стержни в нужный момент ударяют по бумаге через красящую ленту, что обеспечивает формирование на бумаге символов и изображений.

2) Лазерные принтеры.

В лазерном принтере изображение создается лазерным лучом. Им управляет сложная электроника, которая позиционирует луч с высокой точностью. Луч рисует изображение на барабане внутри лазерного принтера. Там, где луч касается барабана, возникает электрический заряд. Барабан вращается над картриджем с тонером, собирая тонер на электрически заряженные участки. Когда барабан входит в контакт с бумагой, тонерное изображение переводится на бумагу. Чтобы закрепить тонер на бумаге, используется тепло. Качество печати приближается к типографскому.

3) Струйные принтеры.

В этих принтерах изображение формируется микрокаплями специальных чернил, выдуваемых на бумагу с помощью сопел. Этот способ печати обеспечивает более высокое качество печати по сравнению с матричными принтерами, он позволяет получить на специальной бумаге высококачественное цветное изображение.

**Графопостроители (плоттеры)** — для вывода графической информации (графиков, чертежей, рисунков) из ПК на бумажный носитель.

**Монитор** - это устройство визуального представления данных. Его основными потребительскими параметрами являются: размер, шаг маски экрана, максимальная частота регенерации изображения, класс защиты.

*3. Средства связи и телекоммуникации*

**Модем** - коммуникационное устройство, позволяющее передавать цифровые данные по аналоговой телефонной линии. Он осуществляет преобразование данных с компьютера в последовательность дискретных (разнотипных) сигналов и их отправку по аналоговой телефонной линии. На другом конце они расшифровываются принимающим модемом путем аналого-цифрового преобразования.

**Факс-модем** - это устройство, сочетающее возможности модема и средства для обмена факсимильными изображениями с другими факс-модемами и обычными телефаксными аппаратами.

**4. Устройства речевого ввода-вывода**

Устройства речевого ввода — это различные микрофонные акустические системы, "звуковые мыши", например, со сложным программным обеспечением, позволяющим распознавать произносимые человеком буквы и слова, идентифицировать их и закодировать.

Устройства речевого вывода — это различные синтезаторы звука, выполняющие преобразование цифровых кодов в буквы и слова, воспроизводимые через громкоговорители (динамики) или звуковые колонки, подсоединенные к компьютеру.

Средства мультимедиа — это комплект аппаратных и программных средств, позволяющих человеку общаться с компьютером, используя самые разные, естественные для себя среды: звук, видео, графику, тексты, анимацию и др. К средствам мультимедиа относятся устройства речевого ввода и вывода информации; высококачественные видео- и звуковые платы, платы видеозахвата, снимающие изображение с видеомагнитофона или видеокамеры и вводящие его в ПК; высококачественные акустические и видеовоспроизводящие системы с усилителями, звуковыми колонками, большими видеоэкранами, а так же внешние запоминающие устройства большой емкости на оптических дисках, часто используемые для записи звуковой и видеоинформации.

### ЛЕКЦИЯ №3

**Тема: Программное обеспечение ПК.**

#### **Классификация программного обеспечения (ПО)**

**Программное обеспечение** - совокупность программ обработки данных.

**Программа** - упорядоченная последовательность команд (инструкций) компьютера для решения задачи.

**Программный продукт** - комплекс взаимосвязанных программ для решения определенной задачи массового спроса, подготовленный к реализации как любой вид промышленной продукции.

**Интегрированное программное обеспечение** — набор нескольких программных продуктов, функционально дополняющих друг друга, поддерживающих единые информационные технологии, реализованные на общей вычислительной и операционной платформе.

Программное обеспечение можно классифицировать по разным признакам. Рассмотрим классификацию по сфере (области) использования программного обеспечения:



### Системное программное обеспечение

**Системное программное обеспечение** - совокупность программ и программных комплексов для обеспечения работы компьютера и компьютерных сетей.



**Базовое программное обеспечение** - минимальный набор программных средств, обеспечивающих работу компьютера.

**Сервисное программное обеспечение (программы-утилиты)** — программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя.

### Базовое программное обеспечение

**1. Операционная система** - это совокупность программ, обеспечивающих управления выполнением пользовательских программ, планирования и управления вычислительными ресурсами ЭВМ.

*Основная функция* всех операционных систем - посредническая. Она заключается в обеспечении нескольких видов взаимодействия (интерфейса):

- *интерфейс пользователя* (взаимодействие между пользователем и программно-аппаратными средствами компьютера);
- *аппаратно-программный интерфейс* (взаимодействие между программным и аппаратным обеспечением);
- *программный интерфейс* (взаимодействие между разными видами программного обеспечения).

Программы, работающие под управлением операционной системы, называют **приложениями**.

Операционные системы для персональных компьютеров можно классифицировать по нескольким признакам:

1) *По реализации интерфейса.*

Различают неграфические (текстовые) и графические операционные системы. *Неграфические* операционные системы (MS DOS) реализуют интерфейс командной строки. Основным устройством управления в данном случае является клавиатура.

*Графические* операционные системы реализуют более сложный тип интерфейс, в котором в качестве органов управления могут выступать клавиатура, мышь или другие манипуляторы. Работа в графической операционной системе основана на взаимодействии активных и пассивных экранных элементов управления. В качестве активного элемента управления выступает указатель мыши, в качестве пассивных элементов управления выступают экранные кнопки, значки, переключатели, флажки, списки, меню и многое другое.

2) *Поддержка многозадачности.*

Однозадачные операционные системы (Ms-Dos) передают все ресурсы вычислительной системы одному исполняемому приложению.

Многозадачность допускает параллельное выполнения нескольких приложений. Различают *вытесняющую* и *невытесняющую* многозадачность.

При *невытесняющей многозадачности* (NetWare) активный процесс выполняется до тех пор, пока он сам по собственной инициативе, не отдаст управление операционной системе для того, чтобы та выбрала из очереди другой готовый к выполнению процесс.

При *вытесняющей многозадачности*\_(Windows 95/98/2000, Windows NT, OS/2, Unix) решение о переключении процесса с одного на другой принимается операционной системой, а не самим активным процессом.

3) *Поддержка многопользовательского режима.*

В зависимости от числа пользователей, одновременно работающих с ОС, ОС делятся на *однопользовательские* (Ms DOS, ранние версии OS/2) и *многопользовательские*\_(Unix, Windows NT). Многопользовательские ОС позволяют нескольким пользователям разделять вычислительные ресурсы одного компьютера.

4) *Поддержка переносимости* (непереносимые и переносимые на другие типы компьютеров);

5) *Поддержка работы в сетях.*

**2. Операционные оболочки** - специальные программы, предназначенные для облегчения общения пользователя с командами операционной системы.

Операционные оболочки имеют неграфический (текстовый) и графический варианты интерфейса пользователя. Эти программы существенно упрощают задание управляющей информации для выполнения команд операционной системы, уменьшают напряженность и сложность работа конечного пользователя.

**3. Сетевые операционные системы** – комплекс программ, обеспечивающий обработку, передачу и хранение данных в сетях.

### **Сервисное программное обеспечение**

**1. Программы диагностики работоспособности компьютера и обслуживания дисков.**

1) Форматирование диска – разбиение диска на дорожки и сектора.

2) Дефрагментация диска (Disk Defragments) – процедура по переносу информации из одних кластеров в другие, в результате которой доступ к любой информации будет более быстрым.

3) Проверка диска (Scandisk) – программа обнаружения ошибок, связанных со сбоями в процессе записи на диск.

4) Восстановление удаленных файлов.

5) Очистка диска.

6) Сведения о ресурсах.

**2. Программы архивирования данных.**

Предназначены для архивации, упаковки файлов путем сжатия хранимой в них информации.

**Архивный файл** – это специальным образом организованный файл, содержащий в себе один или несколько файлов в сжатом или несжатом виде и служебную информацию об именах файлов, дате и времени их создания или модификации, размерах и т. п.

**Архивация** (упаковка) – помещение (загрузка) исходных файлов в архивный файл в сжатом или несжатом виде.

*Цели сжатия файлов:*

- обеспечение более компактного размещения информации на диске;
- сокращение времени и стоимости передачи информации по каналам связи в сетях;
- упрощение переноса файлов с одного диска на другой;
- защита информации от несанкционированного доступа;
- защита от заражения вирусами.

**Разархивация** (распаковка) – процесс восстановления файлов из архива точно в таком виде, какой они имели до загрузки в архив. При распаковке файлы извлекаются из архива и помещаются на диск или в оперативную память.

Наиболее распространенные программы – архиваторы: *WinRAR* и *WinZIP*.

### **3. Антивирусные программы.**

**Компьютерным вирусом** называется специально написанная программа, способная самопроизвольно присоединяться к другим программам, создавая свои копии и внедряя их в файлы, системные области компьютера и в вычислительные сети с целью нарушения работы программ, порчи файлов и каталогов, создания всевозможных помех в работе на компьютере.

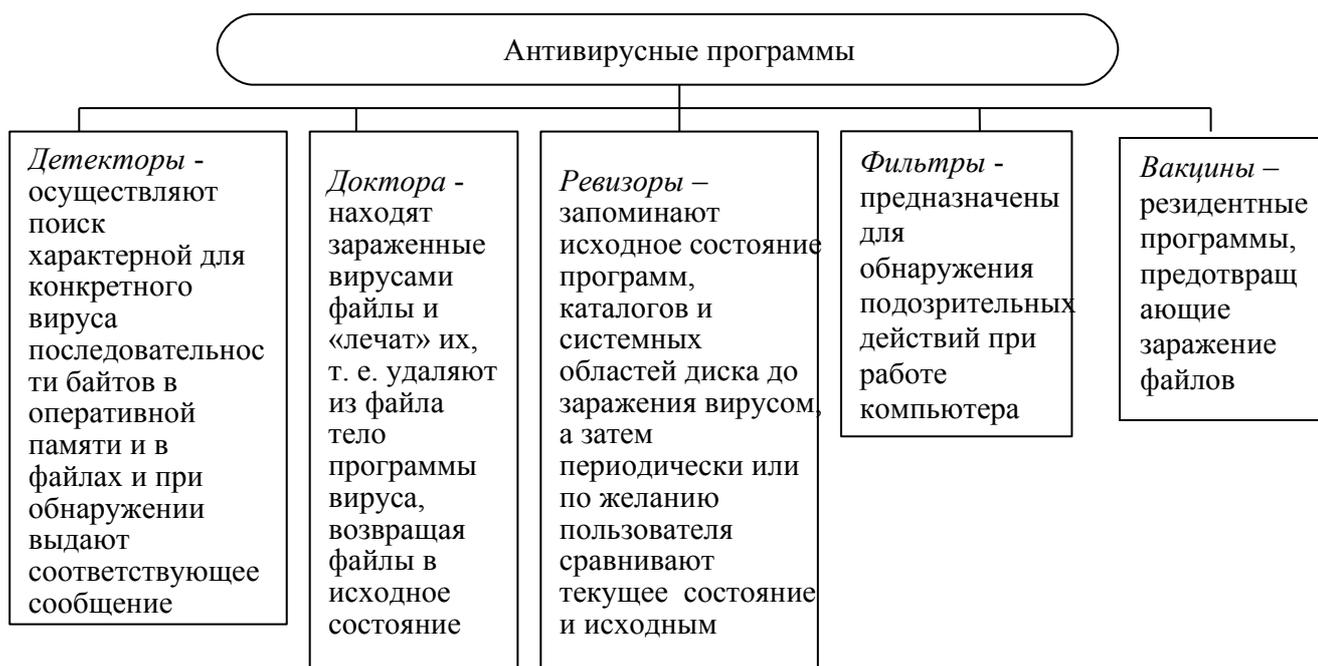
*Признаки появления вирусов:*

- прекращение работы или неправильная работа программы;
- невозможность загрузки операционной системы;
- увеличение размера файлов;
- исчезновение файлов и каталогов или искажение их содержимого;
- увеличение размера свободной памяти;
- подача непредусмотренных звуковых сигналов.





**Антивирусная программа** – это программа, предназначенная для обнаружения, удаления и защиты от компьютерных вирусов.



### Прикладное программное обеспечение

Данный класс программных средств наиболее представлен, что обусловлено прежде всего широким применением средств компьютерной техники во всех сферах деятельности человека, созданием автоматизированных информационных систем различных предметных областей.

#### 1. Текстовые редакторы.

Основные функции *текстовых редакторов* заключаются в автоматизации ввода и редактирования текстовых данных.

В отличие от текстовых редакторов, *текстовые процессоры* позволяют не только вводить и редактировать текст, но и форматировать его, т.е. оформлять. Соответственно, к основным средствам текстовых процессоров относятся средства обеспечения взаимодействия текста, графики, таблиц и других объектов.

Назначение *настольных издательских систем* состоит в автоматизации процесса верстки полиграфических изданий.

#### 2. Графические редакторы.

**Компьютерная графика** – раздел информатики, предметом которого является работа на компьютере с графическими изображениями (рисунками, чертежами, фотографиями и т. д.).

**Графический редактор** – прикладная программа, предназначена для создания и обработки графических изображений на компьютере.

## Типы компьютерной графики

**1. Растровая графика** – графика, состоящая из прямоугольной сетки точек, называемой растром.

Каждая точка растра (пиксель) представлена единственным параметром – цветом.

*Пиксель* - это цветное пятно, которое может принимать различные оттенки. Любое изображение вне зависимости от его сложности - это всего лишь совокупность пикселей.

Редактирование растровых изображений – изменение цветов их отдельных пикселей.

### *Характеристики растровых изображений*

✎ Получаются с помощью сканеров из фотографических изображений, с помощью цифровых камер.

✎ Обеспечивают максимальную реалистичность изображения.

✎ Занимают много памяти.

✎ Качество зависит от размера.

✎ Основные параметры растровых изображений

✎ Количество цветов

✎ Разрешение – количество пикселей на единицу длины (как правило на дюйм)

**2. Векторная графика** состоит из контуров. Контурные представляют собой кривые, имеющие точное математическое описание.

### *Характеристики векторного изображения*

✎ Векторные изображения требуют ручного ввода (построения и рисования). Могут быть получены из растровых с помощью программ трассировки.

✎ Не обеспечивают близкую к оригиналу реалистичность.

✎ Компактны, так как хранят только математическое описание объектов.

✎ Качество изображения не зависит от размера.

✎ Легко редактировать.

Macromedia Flash MX, дает возможность импортировать векторную и растровую графику, преобразовывать растровую в векторную.

**3. Фрактальная графика** – предназначена для автоматической генерации изображений путем математических расчетов. Создание фрактальной художественной композиции состоит не в рисовании или оформлении, а в программировании. Изображение строится по уравнению.

Простейшим фрактальным объектом является фрактальный треугольник.

Данный вид графики часто используют в развлекательных программах.

## Теория цвета

Основные - это три цвета, смешением которых в разных пропорциях можно получить любой цвет.

Дополнительные цвета при смешении дают в глазу ощущение белого цвета, например сине-зеленый и красный, оранжевый и синий, зелено-желтый и фиолетовый цвета.

Для описания цвета используются математические модели, работа которых основана на смешения основных цветов для получения всех других цветов

✍ **RGB** (Red-красный, Green-зеленый, Blue -синий). В этой модели отображает экран монитора, кодирует изображение сканер.

✍ **СМУК** (Cyan-голубой, Magenta-пурпурный, Yellow-желтый, Black). Используется для создания изображений, предназначенных для печати.

### 3. Системы управления базами данных - СУБД.

*Базами данных* называются огромные массивы информации (данных) в определенной предметной области.

### 4. Электронные таблицы.

*Электронная таблица (ЭТ)* – компьютерный эквивалент обычной таблицы, в клетках (ячейках) которой записаны данные различных типов: тексты, даты, формулы, числа.

Для управления ЭТ используется специальный комплекс программ – *табличный процессор*.

Рабочая область ЭТ состоит из *строк* и *столбцов*, имеющих свои имена. Имена строк – это их номера. Нумерация строк начинается с 1 и заканчивается максимальным числом, установленным для данной программы. Имена столбцов – это буквы латинского алфавита сначала от А до Z, затем от AA до AZ, BA до BZ и т. д.

Пересечение строки и столбца образует *ячейку таблицы*, имеющую свой уникальный адрес. Для указания адресов ячеек в формулах используются *ссылки* (например, A2 или C4).

В ЭТ существует понятие *блока* (диапазона) *ячеек*. *Блок ячеек* – группа последовательных ячеек, может состоять из одной ячейки, строки (или ее части), столбца (или его части), а также последовательности строк или столбцов (или их частей). Адрес блока ячеек задается указанием ссылок первой и последней его ячеек, между которыми ставится разделительный символ – двоеточие <:> или две точки подряд <..>.

*Пример.* G3, A1:H1, B1:B10, D4:F5.

Каждая команда ЭТ требует указания блока (диапазона) ячеек, в отношении которых она должна быть выполнена.

Блок используемых ячеек может быть указан двумя путями: либо *непосредственным набором* с клавиатуры начального и конечного адресов ячеек, формирующих диапазон, либо

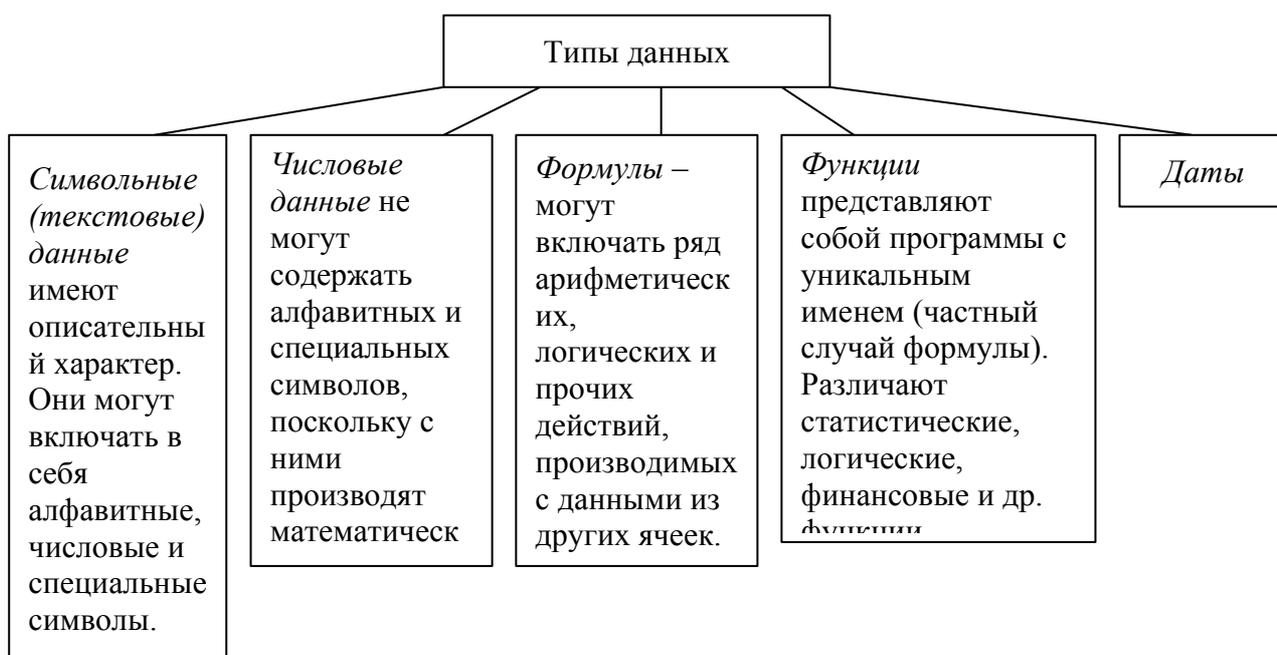
выделением соответствующей части таблицы при помощи клавиш управления курсором.

Удобнее задавать диапазон выделением ячеек.

*Текущей (активной)* называется ячейка ЭТ, в которой в данный момент находится курсор. Адрес и содержимое текущей ячейки выводятся в строке ввода ЭТ.

Каждый документ представляет собой набор таблиц – *рабочую книгу*, которая состоит из одного или многих *рабочих листов*.

В каждую ячейку ЭТ пользователь может ввести данные одного из следующих типов:



В рамках одной и той же ЭТ можно использовать различные форматы представления числовых данных.

Формулы – это выражения, состоящие из числовых величин, адресов ячеек и функций, соединенных знаками арифметических операций. Написание формулы начинается со знака равенства (=).

*Пример.* =(A1+B1)/C2.

В ЭТ могут быть представлены следующие виды функций: математические, статистические, текстовые, логические, финансовые и др.

При копировании или перемещении формулы в другое место таблицы необходимо организовать управление формированием адресов исходных данных. Поэтому в ЭТ при написании формул используются понятия относительной и абсолютной ссылок.

*Абсолютная ссылка* – это не изменяющийся при копировании и перемещении формулы адрес ячейки, содержащий исходное данные.

Для указания абсолютной адресации вводится символ \$. Различают два типа абсолютной ссылки: полная и частичная.

*Полная абсолютная ссылка* указывается, если при копировании или перемещении адрес клетки, содержащий исходное данные, не меняется. Для этого символ \$ ставится перед наименованием столбца и номером строки.

*Пример.* \$B\$5; \$D\$12

*Частичная абсолютная ссылка* указывается, если при копировании или перемещении не меняется номер строки или наименование столбца. При этом символ \$ ставится перед номером строки, а во втором случае – перед наименованием столбца.

*Пример.* B\$5; D\$12; \$B5; \$D12

При абсолютной адресации копируемая формула не изменяется.

*Относительная ссылка* – это изменяющийся при копировании и перемещении формулы адрес ячейки, содержащий исходное данные. Изменение адреса происходит по *правилу относительной ориентации* клетки с исходной формулой и клеток с данными.

*Пример:* В ячейку C1 занесена формула =A\$1+\$B2. Перенести эту формулу в ячейки D2, E4.

	A	B	C	D	E
1			=A\$1+\$B2		
2				=B\$1+\$B3	
3					
4					=C\$1+\$B5

Функция представляет собой программу с уникальным именем, для которой пользователь должен задать конкретные значения аргументов, стоящих в скобках после ее имени.

Функция имеет синтаксис написания: начинается с указания имени функции, затем вводится открывающаяся скобка, указываются аргументы, отделяющиеся «;», а затем – закрывающаяся скобка.

Рассмотрим синтаксис и примеры задания наиболее используемых функций.

### 1. Логические функции.

Логические функции предназначены для проверки выполнения условия или для проверки нескольких условий.

- ✓ **ЕСЛИ** используется для условной проверки значений и формул. Возвращает одно значение, если заданное условие при вычислении дает значение ИСТИНА, и другое значение, если ЛОЖЬ.

Синтаксис: ЕСЛИ(лог\_выражение;значение\_если\_истина;значение\_если\_ложь)

*Пример.* ЕСЛИ(A10>B1;B5/B1;10)

### 2. Статистические функции.

Синтаксические функции позволяют выполнять статистический анализ диапазонов данных.

✓ **СРЗНАЧ** возвращает среднее арифметическое значение своих аргументов.

Синтаксис: СРЗНАЧ(число1;число2;...),

где число1, число2, ... - это от 1 до 30 аргументов, для которых вычисляется среднее значение.

Пример. Если ячейки A1:A5 содержат числа 10, 7, 9, 27 и 2, то СРЗНАЧ(A1:A5) равняется 11.

✓ **СЧЕТ** подсчитывает количество чисел в списке аргументов, используется для получения количества числовых ячеек в интервалах или массивах ячеек.

Синтаксис: СЧЕТ(значение1;значение2;...).

Пример. Если ячейки A1:A7 содержат числа, то СЧЕТ(A1:A7) возвращает число 7.

✓ **МАКС** возвращает наибольшее значение из набора значений.

Синтаксис: МАКС(число1;число2;...).

Пример. Если ячейки A1:A5 содержат числа 10, 7, 9, 27 и 2, то МАКС(A1:A5) равняется 27.

✓ **МИН** возвращает наименьшее значение из набора значений.

Синтаксис: МИН(число1;число2;...).

Пример. Если ячейки A1:A5 содержат числа 10, 7, 9, 27 и 2, то МИН(A1:A5) равняется 2.

### 3. Арифметические и тригонометрические функции.

Арифметические и тригонометрические функции позволяют производить простые и сложные математические вычисления.

✓ **СУММ** суммирует все числа в интервале ячеек.

Синтаксис: СУММ(число1;число2;...).

Пример. Если ячейки A2:E5 содержат числа 5, 15, 30, 40 и 50, то СУММ(A2:E5) равняется 50.

Пример. Даны три числа. Если произведение первого и второго больше суммы всех чисел, то вывести натуральный логарифм первого числа, иначе экспоненту произведения этих чисел.

В ячейке B3 вводится формула для вычисления:

=ЕСЛИ(ПРОИЗВЕД(A1;B1)>СУММ(A1:C1);LN(ABS(A1));EXP(ПРОИЗВЕД(A1:C1)))

Excel располагает рядом команд для построения различных типов диаграмм: круговая диаграмма, линейный график и т. д.

## 5. Системы автоматического проектирования.

Программы этого класса предназначены для автоматизации проектно-конструкторских работ, связанных с разработкой чертежей, схем, диаграмм, графическим моделированием и конструированием, созданием библиотеки стандартных элементов чертежей и их многократным использованием, созданием демонстрационных иллюстраций и мультфильмов.

#### **6. Системы автоматизированного управления.**

Системы автоматизированного управления - самый представительный класс программных продуктов, включающий в себя программные продукты:

- автоматизированного бухгалтерского учета;
- управления финансовой деятельности;
- управления персоналом (кадровый учет);
- управления материальными запасами;
- управления производством;
- банковские информационные системы и т.п.

#### **7. Офисные приложения.**

Данный класс программных продуктов охватывает программы, обеспечивающие организационное управление деятельностью офиса: элементарные текстовые редакторы; органайзеры; автоматизированный перевод документов; средства проверки орфографии и распознавания текста; электронная почта.

#### **8. Экспертные системы.**

Предназначены для анализа данных, содержащихся в банках знаний, и выдачи рекомендаций по запросу пользователя. Такие системы применяют в тех случаях, когда исходные данные хорошо формализуются, но для принятия решения требуются обширные специальные знания. Характерными областями использования экспертных систем являются юриспруденция, медицина, фармакология, химия.

Характерной особенностью экспертных систем является их способность к *саморазвитию*. Исходные данные хранятся в базе знаний в виде фактов, между которыми с помощью специалистов-экспертов устанавливается определенная система отношений. Если на этапе тестирования экспертной системы устанавливается, что она дает некорректные рекомендации и заключения по конкретным вопросам, или не может дать их вообще, это означает либо отсутствие важных фактов в ее базе, либо нарушения в логической системе отношений. И в том и в другом случае экспертная система сама может сгенерировать достаточный набор запросов к эксперту и автоматически повысить свое качество.

#### **9. Редакторы HTML (Web- редакторы).**

Это особый класс редакторов, объединяющий в себе свойства текстовых и графических редакторов. Они предназначены для создания и редактирования так называемых Web-документов (Web- страниц Интернета). Программы этого класса эффективно используют для подготовки электронных документов и мультимедийных изданий.

#### **10. Геоинформационные системы (ГИС)**

ГИС предназначены для автоматизации картографических и геодезических работ на основе информации, полученной топографическими или аэрокосмическими методами.

#### **11. Автоматизация научно-исследовательских работ.**

Программы, автоматизирующие научно-исследовательские работы, используют вычислительную мощь компьютера для решения расчетных задач.

К программам автоматизации научно-исследовательских работ относятся математические, статистические пакеты и др.

#### **12. Программные средства мультимедиа.**

Этот класс программных продуктов является относительно новым. Основное назначение программных продуктов мультимедиа - создание и использование аудио- и видеоинформации для расширения информационного пространства пользователя.

#### **13. Системы видеомонтажа.**

Системы видеомонтажа предназначены для цифровой обработки видеоматериалов, их монтажа, создания видеоэффектов, устранения дефектов, наложения звука, титров и субтитров.

#### **14. Компьютерная обработка звука. Музыкальные редакторы.**

Компьютерная обработка звука ориентирована на систему цифровой записи. Музыкальные редакторы позволяют обрабатывать звук, записанный на звуковой носитель; реставрировать старые записи с помощью встраиваемых приложений; осуществлять нотную запись; создавать многоканальную запись; подготавливать звуковые файлы к публикации в Интернет.

#### **15. Обучающие программы.**

Обучающие программы предназначены для самостоятельного изучения данной темы, широко используются в заочном или дистанционном образовании.

### **Инструментарий технологии программирования**

Это совокупность программ и программных комплексов, обеспечивающих технологию разработки, отладки и внедрения создаваемых программных продуктов.

Инструментарий технологии программирования обеспечивает процесс разработки программ и включает специализированные программные продукты, которые являются инструментальными средствами разработчика. Программные продукты данного класса

поддерживают все технологические этапы процесса проектирования, программирования (кодирования), отладки и тестирования создаваемых программ. Пользователями технологии программирования являются системные и прикладные программисты.

Инструментарий технологии программирования включают в себя следующие классы:

- средства для создания приложений (языки и системы программирования);
- средства для создания информационных систем (CASE - технологии).

**Языки программирования** - формальный язык для описания алгоритма решения задачи на компьютере.

**Системы программирования** - хорошо интегрированная система, включающая в себя как минимум:

- специализированный текстовый редактор;
- компилятор для перевода текста программы в машинный код;
- библиотека функций;
- редактор связей для связывания модулей (файлов с исходными текстами) и стандартных функций, находящихся в библиотеках;
- исполнимый код - это законченная программа с расширением .COM или .EXE, которую можно запустить на любом компьютере, где установлена операционная система, для которой эта программа создавалась;
- справочную систему;
- отладчик, который позволяет анализировать работу программы во время ее выполнения по шагам.

**CASE-технологии** - программный комплекс, автоматизирующий весь технологический процесс анализа, проектирования, разработки и сопровождения сложных программных систем.

## ЛЕКЦИЯ №4

**Тема: Файловая структура. Операционная система Ms-Dos. Norton-подобные оболочки.**

### *Файловая система*

Файловая система (ФС) является важной частью любой операционной системы, которая отвечает за организацию хранения и доступа к информации на каких-либо носителях.

**ФС** - это совокупность именованных наборов данных и программ на внешних носителях, структуру и организацию которых поддерживает операционная система.

В широком смысле понятие ФС включает:

- ✓ совокупность всех файлов на диске,

- ✓ наборы служебных структур данных, используемых для управления файлами, такие как, например, каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске,
- ✓ комплекс системных программных средств, реализующих управление файлами, в частности операции по созданию, уничтожению, чтению, записи, именованию файлов, установке атрибутов и уровней доступа, поиску и т.д.

Различие между файловыми системами заключается, в основном, в способах распределения пространства между файлами на диске и организации на диске служебных областей.

**Файл** - это поименованная совокупность элементов информации хранящаяся на магнитных носителях - дисках, лентах.

Имена файлов записываются следующим образом: ИМЯ . ТИП, где ИМЯ набор символов (короткое не более 8 символов, длинное – 256 ) латинского алфавита, цифр и специальных символов ~ & @ ( ) % { } \_ # \$, а ТИП или РАСШИРЕНИЕ файла состоит из не более чем 3 символов. В отличие от имени тип может отсутствовать в спецификации файла.

Тип файла используется для классификации, определения принадлежности к какой-то группе с общими свойствами.

Тип	Назначение
ARJ	Архивный файл
BAT	Командный файл
COM	Командный системный файл, исполняемый файл
DAT	Файл данных
DOC	Файл документов (текстовый)
EXE	Исполняемый файл
PAS	Программа на языке Паскаль
SYS	Файлы, расширяющие возможности операционной системы
TXT	Текстовый файл

При использовании имен файлов в качестве параметров команд ОС необходимо указывать адрес или путь к файлу. Путь к файлу называется цепочка символов, начиная с имени дисководов, корневого каталога и последующих подкаталогов вплоть до каталога, содержащего необходимый файл.

**Имя дисководов** - это одна из букв латинского алфавита. Персональный компьютер имеет несколько накопителей на магнитных носителях, исходя из этого, принято обозначать А: и В: - гибкие диски, С: D: и т.д. - жесткие диски.

**Каталог** - это справочник файлов и библиотек со ссылками на их расположение, содержащее информацию о файлах (имя, тип, размеры в байтах, дата и время создания, атрибуты) и других каталогах, называемых подкаталогами, используется операционной системой для определения местоположения файла.

На каждом диске имеется один главный или *корневой* каталог. Каталоги, входящие в корневой каталог называются *подкаталогами 1-го уровня*. Каталоги, входящие в состав подкаталога 1-го уровня называются *подкаталогами 2-го уровня* и т.д. Каждый подкаталог является оглавлением, содержащим перечень имен файлов и подкаталогов, возможны варианты, когда в оглавлении имеются только имена файлов.

### **Организация файловой системы**

Принцип организации файловой системы — табличный. Поверхность жесткого диска рассматривается как трехмерная матрица, измерениями которой являются номера *поверхности, цилиндра* и *сектора*. Под цилиндром понимается совокупность всех дорожек, принадлежащих разным поверхностям и находящихся на равном удалении от оси вращения. Данные о том, в каком месте диска записан тот или иной файл, хранятся в системной области диска в специальных *таблицах размещения файлов* (FAT-таблицах). Поскольку нарушение FAT - таблицы приводит к невозможности воспользоваться данными, записанными на диске, к ней предъявляются особые требования надежности, и она существует в двух экземплярах, идентичность которых регулярно контролируется средствами операционной системы.

*Наименьшей физической единицей хранения данных является сектор*. Размер сектора равен 512 байт. Поскольку размер FAT- таблицы ограничен, то для дисков, размер которых превышает 32 Мбайт, обеспечить адресацию к каждому отдельному сектору не представляется возможным. В связи с этим группы секторов условно объединяются в кластеры. *Кластер является наименьшей единицей адресации к данным*. Размер кластера, в отличие от размера сектора, не фиксирован и зависит от емкости диска.

Операционные системы MS-DOS, OS/2, Windows 95, Windows NT реализуют 16-разрядные поля в таблицах размещения файлов. Такая **ФС** называется **FAT 16**. Она позволяет разместить в FAT-таблицах не более 65 536 записей ( $2^{16}$ ) о местоположении единиц хранения данных и, соответственно, для дисков объемом от 1 до 2 Гбайт длина кластера составляет 32 Кбайт (64 сектора).

Высокопроизводительная **ФС HPFS** (High Performance File System) была представлена фирмой IBM в 1989 году вместе с операционной системой OS/2 1.20. По HPFS позволяет использовать жесткие диски объемом до 2 Терабайт (первоначально до 4 Гбайт). Кроме того, она поддерживает разделы диска размером до 512 Гб и позволяет использовать

имена файлов длиной до 255 символов (на каждый символ при этом отводится 2 байта). В HPFS по сравнению с FAT уменьшено время доступа к файлам в больших каталогах.

HPFS распределяет пространство на диске не кластерами как в FAT, а физическими секторами по 512 байт, что не позволяет ее использовать на жестких дисках, имеющих другой размер сектора. Эти секторы принято называть блоками. Чтобы уменьшить фрагментацию диска, при распределении пространства под файл HPFS стремится, по возможности, размещать файлы в последовательных смежных секторах. Фрагмент файла, располагающийся в смежных секторах, называется *экстендом*.

Для нумерации единиц распределения пространства диска HPFS использует 32 разряда, что дает  $2^{32}$ , или более 4 миллиардов номеров. Однако HPFS использует числа со знаком, что сокращает число возможных номеров блоков до 2 миллиардов. Помимо стандартных атрибутов файла, HPFS поддерживает *расширенные атрибуты* файла (Extended Attributes, EA), которые могут содержать до 64 Кб различных дополнительных сведений о файле.

Диск HPFS имеет следующие три базовые структуры: загрузочный блок (BootBlock), дополнительный блок (SuperBlock) и резервный блок (SpareBlock).

**ФС VFAT** (Virtual FAT), реализованная в Windows NT 3.5, Windows 95 (DOS 7.0), - это файловая система FAT, включающая поддержку длинных имен файло в кодировке UNICODE (каждый символ имени кодируется 2 байтами). VFAT использует ту же самую схему распределения дискового пространства, что и файловая система FAT, поэтому размер кластера определяется величиной раздела.

**ФС FAT32** - усовершенствованная версия файловой системы VFAT, поддерживающая жесткие диски объемом до 2 терабайт. Впервые файловая система FAT32 была включена в состав ОС Windows 95 OSR 2.

Главные отличия от предыдущих версий FAT состоят в следующем. Блок начальной загрузки на разделах с FAT32 был увеличен до 2 секторов и включает в себя резервную копию загрузочного сектора, что позволяет системе быть более устойчивой к возможным сбоям на диске. Объем, занимаемый таблицей размещения файлов, увеличился, поскольку теперь каждая запись в ней занимает 32 байта, и общее число кластеров на разделе FAT32 больше, чем на разделах FAT. Соответственно, выросло и количество зарезервированных секторов.

**ФС NTFS** (New Technology File System) - наиболее предпочтительная файловая система при работе с ОС Windows NT (Windows 2000 и XP также являются NT системами), поскольку она была специально разработана для данной системы. В состав Windows NT входит утилита convert, осуществляющая конвертирование томов с FAT и HPFS в тома

NTFS. В NTFS значительно расширены возможности по управлению доступом к отдельным файлам и каталогам, введено большое число атрибутов, реализована отказоустойчивость, средства динамического сжатия файлов. NTFS позволяет использовать имена файлов длиной до 255 символов, при этом она использует тот же алгоритм для генерации короткого имени, что и VFAT. NTFS обладает возможностью самостоятельного восстановления в случае сбоя ОС или оборудования, так что дисковый том остается доступным, а структура каталогов не нарушается.

NTFS позволяет хранить файлы размером до 16 эксабайт ( $2^{64}$  байт) и располагает встроенным средством уплотнения файлов в реальном времени.

## **Операционная система Microsoft DOS.**

### **1. Начальные сведения о DOS.**

DOS является основной частью программного обеспечения, она контролирует все, даже другое программное обеспечение или программы, находящиеся в вашем компьютере. DOS означает Disk Operating System (дисксовая операционная система), которая необходима для запуска и работы компьютера.

DOS является однопользовательской и однозадачной ОС. Диалог пользователя с DOS осуществляется в форме команд, которые записываются в командной строке, начинающейся с приглашения DOS. Командная строка обычно имеет следующий вид: C:\>\_

### **2. Составные части DOS**

1. Базовая система ввода/вывода (BIOS), находящаяся в постоянной памяти (постоянном запоминающем устройстве, ПЗУ) компьютера. Эта часть ОС является встроенной в компьютер. Её назначение состоит в выполнении наиболее простых и универсальных услуг ОС, связанных с осуществлением ввода-вывода. Базовая система ввода-вывода содержит так же тест функционирования компьютера, проверяющий работу памяти и устройств компьютера при включении его электропитания. Кроме того, BIOS содержит программу вызова загрузчика ОС.

2. Загрузчик ОС (ядро DOS) - это очень короткая программа, находящаяся в первом секторе каждой дискеты с ОС DOS. Функции этой программы заключаются в считывании в память еще двух модулей ОС, которые и завершают процесс загрузки DOS. На жестком диске (винчестере) загрузчик ОС состоит из двух частей. Это связано с тем, что жесткий диск может быть разбит на несколько логических дисков. Первая часть загрузчика находится в первом секторе жесткого диска, она выбирает, с какого логического диска следует продолжить загрузку. Вторая часть загрузчика находится в первом секторе этого диска, она считывает в память модули DOS и передает им управление.

3. Дисковые файлы Io.sys и Msdos.sys. В файле Io.sys содержится BIOS, т.е. система, которая (в дополнении к BIOS в ПЗУ) управляет внутренними процессами в компьютере и контролирует их. В определенном смысле она устанавливает связь между отдельными составными частями компьютера, между его кристаллами и другими электронными микросхемами. Файл Msdos.sys обеспечивает связь со стандартными системными устройствами ПК. Этот файл предоставляет системному программисту возможность использовать процедуры, входящие в состав ОС, а не писать каждый раз для этого специальные программы.

4. Командный процессор находится в файле *Command.com* на диске, с которого загружается ОС. Командный процессор DOS обрабатывает команды, вводимые пользователем. Если команда соответствует синтаксису ОС, то она выполняется.

Команды в свою очередь делятся на внутренние и внешние. Все команды, которые выполняются непосредственно программой *Command.com*, представляют собой внутренние команды, их можно в любое время вызвать в командную строку. Внутренние команды наиболее часто используются и они необходимы для обеспечения минимальных потребностей управления компьютером. Поэтому они постоянно (резидентно) находятся в оперативной памяти. Все остальные команды DOS, которые не входят в число программ, содержащихся в системном файле *Command.com*, называются внешними командами DOS.

5. Драйверы устройств - это специальные программы, которые дополняют систему ввод/вывода DOS и обеспечивают обслуживание новых или нестандартное использование имеющихся устройств.

### **3. Основные команды DOS**

Диалог пользователя с DOS осуществляется в форме команд. Каждая команда означает, что DOS должна выполнить те или иные действия. Команда DOS состоит из имени команды и, возможно, параметров, разделенных пробелами. Имя команды DOS и параметры могут набираться как прописными, так и строчными латинскими буквами. Ввод каждой команды заканчивается нажатием клавиши <ENTER>.

**1) Создание файла: сору соп имя файла  
текст файла**

В конце каждой строки нажимается клавиша **ENTER**, в конце последней нажимается клавиша **F6** или сочетание клавиш **Ctrl + Z**. Затем, после нажатия клавиши **ENTER**, файл запишется на диск под указанным именем.

**2) Переименование файлов:**

**ren диск\путь\имя файла1 имя файла2**

Первое имя файла в команде задает имя переименовываемого файла, второе - новое имя файла.

### 3) Копирование файлов:

**copy имя файла1 имя файла2**

Первое имя файла в команде задает имя копируемого файла, второе имя файла, в который копируется информация из первого файла. Если файл2 уже существует на диске, то при копировании информация файла1 накладывается на информацию файла2 (т.е. предыдущая информация стирается). Если же файла2 не существовало ранее на диске, то при копировании автоматически создается файл2 с такой же информацией, как в файле1.

### 4) Удаление файлов:

**del дисковод:путь\имя файла**

Для переименования, копирования и удаления группы файлов используются шаблоны, т.е. символы "\*" и "?".

Например:

`del *.txt` удалить все файлы с расширением .txt

`ren *.doc *.txt` переименовать файлы с расширением .doc в файлы с расширением .txt

### 5) Просмотр содержимого файла:

**type имя файла**

Примечание: во всех командах для работы с файлами можно использовать полное имя файла. Если полное имя не указывается, то все действия будут производиться в текущем (рабочем) каталоге (т.е. имя диска и имя каталога можно опустить).

### 6) Изменение текущего каталога:

**cd дисковод:путь**

Например:

`cd\` - переход в корневой каталог текущего диска;

`cd..` - подъем на один уровень дерева каталогов.

### 7) Вывод оглавления каталога:

**dir дисковод:путь\имя файла (/p) (/w)**

Параметр /p - постраничный вывод оглавления. При указании этого параметра DOS после заполнения экрана будет ждать до тех пор, пока пользователь не нажмет любую клавишу, после чего будет выводиться следующую страницу. В оглавлении каталога выводится полная информация о файлах и каталогах. Параметр /w задает вывод только информации об именах файлов в каталоге. Имена выводятся по пять в каждой строке.

### 8) Создание нового каталога: **md дисковод:путь**

### 9) Удаление каталога: **rd дисковод:путь**

Удалять можно только пустой каталог, т.е. каталог, не содержащий файлов и подкаталогов.

**10) Смена текущего дисковода: дисковод:**

**11) Установка даты: DATE**

- эта команда отображает текущую дату.

Если Вы нажмете клавишу <ENTER>, то эта дата останется неизменной.

Если Вы хотите ввести новую дату, то запишите ее после команды DATE по предложенному образцу.

**12) Установка времени: TIME**

- эта команда выводит на экран дисплея используемое MS-DOS текущее системное время, которое при необходимости можно изменить. Если Вы нажмете клавишу <ENTER>, то время останется неизменным. Если Вы хотите ввести новое время, то запишите его после команды TIME по предложенному образцу.

**Пример работы с командами MS-DOS**

1. На диске C: создать каталог MM.	Md C:\MM ↵
2. В MM создать файл t1.txt.	Copy con C:\MM\t1.txt ↵ информация F6 или Ctrl+Z ↵
3. Просмотреть каталог MM.	Dir C:\MM ↵
4. Просмотреть файл t1.txt.	Type C:\MM\t1.txt ↵
5. Перейти в каталог MM.	Cd C:\MM ↵
6. Просмотреть файл t1.txt.	Type t1.txt ↵
7. Перейти на диск C:.	Cd \ ↵
8. Файл t1.txt скопировать в BP\BIN.	Copy C:\MM\t1.txt C:\BP\BIN ↵
9. Переименовать файл t1.txt в BP\BIN в файл m10.txt.	Ren C:\BP\BIN\t1.txt m10.txt ↵
10. Перейти в каталог BIN.	Cd C:\BP\BIN ↵
11. Просмотреть каталог BIN.	Dir ↵
12. Перейти на диск C:.	Cd \ ↵
13. Удалить MM и m10.txt.	Del C:\BP\BIN\m10.txt ↵ Rd C:\MM ↵

## Операционные оболочки. Norton Commander.

**Программа-оболочка** – это программа, один из модулей которой, называемый резидентным, постоянно находится в оперативной памяти компьютера.

Оболочка позволяет эффективно работать с файловой системой дисков, а также запускать программы на исполнение.

### Элементы экрана Windows Commander.

Главное меню	
C:\директория	C:\директория
..	..
Левая панель	Правая панель
Информационная строка	Информационная строка
Строка приглашения Ms-Dos	
Меню функциональных клавиш	

**Главное меню** расположено в верхней строке окна и включает в себе все возможности программы, рассортированные по более или менее однотипным командам. Для входа в меню необходимо либо воспользоваться мышью, либо нажать F10, затем перемещаться по меню с помощью клавиш управления курсором. Покинуть меню можно, нажав клавишу Esc.

**Панели** – две равноправные таблицы, занимающие основную часть окна. Внутри панелей может отображаться различная информация. Одна из панелей является активной (на ней находится курсор и заголовок панели подсвечен), другая панель – пассивная. Смена активной панели достигается нажатием Tab или щелчком мыши в любой части панели. При этом курсор перемещается с одной панели на другую.

**Командная строка** располагается под панелями и служит для выполнения команд MS-DOS.

**Меню функциональных клавиш** расположено в самой нижней строке экрана. Оно имеет вид подсказок к функциональным клавишам.

**Запуск команд MS-DOS из командной строки.** Экран выполнения команд раскрывается в отдельном окне. Выполняемые команды запоминаются Windows Commander. Следовательно, их можно повторить. Комбинация клавиш Ctrl+E позволяет перелистать команды по одной в командной строке в порядке, обратном их выполнению. Для выбора одной из выполненных команд можно также воспользоваться журналом команд, к появлению которого приводит также нажатие Alt+F8.

**Панели** предназначены для представления различной информации на экране. С панелями связаны пункт меню Вид, который имеет три основных раздела. Первый раздел меню Вид содержит команды, управляющие тем, что выводится в панелях, второй раздел – как выводится, третий раздел — режимы сортировки.

**Просмотр содержимого файла.** Для просмотра содержимого файла служит клавиша F3. На экране раскрывается окно с содержимым того файла, на котором стоял курсор. Выход из режима просмотра — Esc.

**Редактирование файла встроенным редактором.** Для редактирования содержимого файла служит клавиша F4. На экране раскрывается окно с содержимым того файла, на котором стоял курсор. Кроме того, содержимое файла можно изменять. При нажатии Shift+F4 сначала появляется диалоговое окно, в котором нужно указать имя того файла, который подлежит редактированию. Если такого файла не существует, то появляется пустое окно для написания содержимого файла. Таким образом, Shift+F4 может служить для создания новых текстовых файлов. Выход из режима редактирования — Esc.

**Копирование файла или каталога.** Для копирования файлов и каталогов служит клавиша F5. Технология копирования заключается в следующем.

- 1) Раскрыть в одной панели тот каталог, в который копируем.
- 2) В другой панели поставить курсор на тот файл или каталог, который копируем.
- 3) Нажать F5.
- 4) Нажать Enter.

Обратите внимание, что при нажатии F5 появляется диалоговое окно копирования. При необходимости в нём можно указать подробности копирования (куда копируем, с каким именем копируем и т.д.).

При использовании комбинации Shift+F5 предлагается ввести имя копируемого файла.

**Переименование и перенос файла или каталога.** Для переноса файлов и каталогов из одного места дерева в другое служит клавиша F6. Технология переноса заключается в следующем.

- 1) Раскрыть в одной панели тот каталог, в который переносим.
- 2) В другой панели поставить курсор на тот файл или каталог, который переносим.

- 3) Нажать F6.
- 4) Нажать Enter.

Таким образом, перенос файлов и каталогов происходит так же, как и копирование.

Для переименования файла необходимо в диалоговом окне переименование/перенос написать новое имя файла вместо пути, который предлагает Windows Commander для переноса. Необходимо подчеркнуть, что переименование и перенос это практически одно и то же, так как полным именем файла считается его имя с указанием пути к нему от корневого каталога диска.

При использовании комбинации Shift+F6 предлагается переименовать файл или каталог.

**Создание нового каталога.** Для создания каталога необходимо перейти в каталог, внутри которого будет создаваться новый каталог и нажать F7. Появится диалоговое окно для ввода имени нового каталога. После нажатия Enter каталог будет создан.

**Удаление файлов и каталогов.** Удаление происходит с помощью клавиши F8, после нажатия которой нужно необходимое количество раз подтвердить удаление.

**Работа с группами файлов и каталогов.** Зачастую возникает необходимость выполнения изученных нами операций со многими файлами и каталогами. Тогда их необходимо предварительно выделить. Выделения и снятие выделения для отдельного файла или каталога происходит с помощью клавиши Insert или правым щелчком мыши. Выбранный файл отмечается другим цветом. В низу панели появляются сведения о количестве выбранных файлов и объеме занимаемой ими дисковой памяти.

Имеется возможность выбрать группу файлов по маске, для этого необходимо нажать на клавишу СЕРЫЙ ПЛЮС и задать маску (образец) для выбора. В маске можно использовать символы \* и ?. Их смысл тот же, что и командах MS DOS. Чтобы отменить выбор группы файлов по маске, нажмите клавишу СЕРЫЙ МИНУС и задайте маску файлов, выбор которых хотите отменить.

**Поиск файлов.** Для поиска файлов на текущем диске используется пункт меню Команды, Поиск файлов или комбинация клавиш Alt+F7. В диалоговом окне «Поиска файла» следует ввести имя файла или шаблон для поиска. После поиска Вы получите список искомых файлов, по которому можно перемещать курсор. Нажатие Enter на одном из найденных файлов приводит к открытию нужной папки с выбранным файлом.

## ЛЕКЦИЯ №5

### Тема: Операционная система Windows-95/98/2000.

#### *Основные характеристики и история создания.*

Первая графическая многооконная операционная оболочка Windows фирмы Microsoft появилась в 1986 г. После своего возникновения она пережила ряд модификаций, но не все из них были удачными. Однако в 1991 г. вышла версия операционной среды Windows 3.1 и несколько позже сетевой вариант Windows 3.11 For WorkGroups, завоевавшие широкое признание миллионов пользователей. Дальнейший ряд Windows-продуктов продолжили высокопроизводительные многозадачные, полнофункциональные операционные системы с графическим интерфейсом Windows 95, Windows 98, Windows 2000 для компьютеров платформы IBM PC.

*Основными характеристическими чертами* операционной системы Windows 95 являются:

1. *Единый графический пользовательский интерфейс*, который составляют рабочий стол, окна, панель задач и другие графические объекты (кнопки, пиктограммы, списки и т.п.).
2. *Интегрированная операционная система*, ядро которой загружается в момент включения компьютера, активизирует графический интерфейс пользователя и обеспечивает полную совместимость с операционной системой MS-DOS.
3. *Объектно-ориентированная система*.
4. *Программная совместимость* обеспечение полной независимости программ от аппаратной части компьютера.
5. *Вытесняющая многозадачность* - свойство операционной системы самостоятельно в зависимости от внутренней ситуации передавать или забирать управление у того или иного приложения, не позволяющее одному приложению занять все аппаратные ресурсы.
6. *32-разрядная операционная система*, поддерживающая 16-разрядные приложения без всякой их модификации.
7. *Многопоточность* - свойство операционной системы выполнять операции одновременно над потоками нескольких 32-битовых приложений. Поток - это некоторая часть процесса, который может быть выделено процессорное время для одновременного выполнения наряду с другими потоками.
8. *Сетевые возможности*. Хотя ОС Windows предназначена для управления автономным компьютером, но также содержит все необходимые средства для создания небольшой локальной одноранговой сети: совместное использование ресурсов файлового сервера, принтеров, факс-модемов. Windows 95 имеет средства для интеграции компьютера во всемирную сеть: использование электронной почты и других средств коммуникации.

9. *Средства обмена данными между приложениями*: буфер обмена (Clipboard), технологии DDE (Dynamic Data Exchange), OLE (Object Linking and Embedding).
10. *Интерфейс мультимедиа*, включает в себя лазерный проигрыватель (CD-плеер), обеспечивает поддержку видеодисков и видеомagneтофонов и т.п.
11. *Поддержка длинных имен* файлов и папок (до 255 символов).
12. *Использование технологии Plug and Play ("включи и работай")* позволяет осуществлять функции распознавания новых устройств для их установки и настройки, при этом обеспечивает динамическое изменение конфигурации системы и автоматического уведомления об этом программных приложений. Технология Plug and Play позволяет работать с устройствами, не подчиняющимися спецификации Plug and Play, упрощая их настройку и управление оборудованием. Для корректного обращения с системными ресурсами компьютера Windows 95 отслеживает все устройства и выделяемые им ресурсы. Диспетчер устройств позволяет получить информацию обо всех найденных системой устройствах и изменить при необходимости их конфигурацию.
13. *Реализация принципа WYSIWYG -What You See Is What You Get ("что видишь, то и получаешь")*. Принцип реализуется при выводе на печать информации, полностью соответствующей изображению на экране.
14. *Технология AutoPlay* позволяет автоматически озвучивать работу с Windows при установленных средствах мультимедиа.
15. *Режим MouseKeys* позволяет все действия с мышью выполнять через клавиатуру).

По сравнению с Windows 95, Windows 98 включает средства, позволяющие компьютеру работать быстрее без добавления нового оборудования. В состав Windows 98 входит ряд программ, совместное применение которых повышает производительность компьютера:

- Служебные программы позволяют быстрее выполнять программы, проверять жесткий диск на наличие ошибок и освобождать место на диске, обеспечивать бесперебойную работу системы.
- Проверка диска запускается автоматически после неверного выключения ОС.
- Новый Web-узел ресурсов Microsoft Windows Update автоматизирует процесс обновления драйверов и системных файлов и обеспечивает новейшие возможности технической поддержки.
- Обозреватель Интернета Internet Explorer делает ряд функций доступными с рабочего стола Windows: каналы Web - узлов на рабочем столе, возможности поиска в Интернет, панели обозревателя. Приложение Internet Explorer объединяет рабочий стол с Web,

благодаря чему рабочий стол и его папки будут выглядеть и действовать так же, как при работе с Web. Такой рабочий стол называется Active Desktop.

Главными новыми технологическими решениями, реализованных в Windows 2000, являются расширение сетевых возможностей и усовершенствование функций защиты информации в сетях. Windows 2000 оснащена целым рядом свойств управления клиентами и серверами, позволяющих снизить общую стоимость эксплуатации операционной системы.

#### *Основные элементы графического интерфейса Windows.*

Задача интерфейса - сделать компьютер доступнее, понятнее для пользователя.

По замыслу разработчиков, общий вид монитора ассоциируется с видом обычного рабочего стола, на котором располагаются документы (окна, подобные листам бумаги).

**Рабочий стол (DeskTop)** - все пространство экрана в среде Windows с расположенными на нем графическими объектами.

Вдоль одной из границ (чаще в нижней части) рабочего стола находится *панель задач*.

Панель задач содержит:

- кнопку *ПУСК* - *главное системное меню*;
- доступ ко всем открытым приложениям. При открытии приложения на панели задач появляется соответствующая открытому окну кнопка. Нажатие этой кнопки позволяет быстро перейти в выбранное окно.
- пиктограммы специальных "фоновых" приложений: часы, индикатор текущей раскладки клавиатуры и др.

**Главное системное меню** предназначено для быстрого запуска программ, поиска файлов, обеспечения доступа к справке, вызов панели управления для настройки компьютера и др. Меню содержит в себе несколько пунктов, при подведении указателя мыши на одном из пунктов автоматически открывается подменю для выбора нужной операции.

1. Пункт *Программа* позволяет запускать программы. Содержит в себе подпункты:
  - *Автозагрузка*, с перечисленными программами и документами, которые будут автоматически открываться сразу после запуска Windows.
  - *Стандартные*, содержащий в себе стандартный набор прикладных программ Windows (служебные программы, игры, элементарные текстовый редактор Блокнот, графический редактор Paint, калькулятор).
2. Пункт *Документы* содержит ярлыки 15 последних открывавшихся документов.
3. Пункт *Избранное* характерен для ActiveDesktop и позволяют осуществлять доступ к средствам всемирной сети.
4. Пункт *Настройка* позволяет настраивать:

- панель задач, в том числе пункты главного меню;
  - панель управления, доступ к которой также можно получить через папку *Мой Компьютер*;
  - свойства папки *Принтеры и др.*
5. Пункт *Найти* позволяет осуществлять поиск папок и файлов на дисках компьютера, в локальной сети, в Интернет. Поиск можно осуществлять, используя маску файлов, дату создания файла, по типу файла, его размеру, поиск по тексту.
  6. Пункт *Справка* выводит справку по работе с Windows.
  7. Пункт *Выполнить* позволяет запускать программы на исполнение, при этом необходимо указать путь и имя запускаемого файла.
  8. Пункт *Завершение работы* необходим для корректного завершения работы с Windows, перезагрузки компьютера, перезагрузки с выходом в режим MS-DOS.

На рабочем столе размещено несколько графических объектов - ярлыков.

**Ярлык** - маленький файл-указатель, с помощью которого можно быстро получить доступ к объекту (файлу, каталогу, диску, программе).

Ярлык представлен в виде значка (**пиктограммы**) и названия. Двойной щелчок по ярлыку открывает объект, с которым он связан.

Основными объектами файловой структуры Windows являются файл, каталог, ярлык. С этими объектами можно проделывать следующие операции: создавать, переименовывать, удалять, копировать, перемещать и другие. В Windows, по аналогии с обычными терминами, принято называть каталог - папкой, файл - документом.

#### **Папка *Мой компьютер*.**

Папка *Мой компьютер* отражает содержание всего компьютера целиком. Папка позволяет просмотреть содержимое находящихся на компьютере дисков, доступ к панели управления, сетевому окружению.

#### **Утилита *Корзина*.**

Корзина - специальная папка, предназначена для временного хранения удаленных файлов, папок, ярлыков. Она позволяет восстановить объекты, удаленные по ошибке. Размер корзины устанавливает пользователь. По мере работы следует регулярно очищать корзину, особенно когда возникают проблемы со свободным дисковым пространством.

#### **Панель управления.**

Утилита *Панель управления* предназначена для конфигурирования и установки параметров Windows: изменения режима работы ОС и пользовательского интерфейса, установки программного и аппаратного обеспечения, настройки рабочих параметров

клавиатуры, мыши, экрана и т.п. Окно папки *Панель управления* может содержать несколько десятков значков.

### **Проводник.**

Проводник - программное средство Windows, предназначенное для управления файловой системой, обеспечивающее доступ к локальным и сетевым ресурсам. Проводник обобщает в себе возможности, заложенные в режимах *Мой компьютер* и *Сетевое окружение*. Он одновременно отображает структуру, находящихся на компьютере папок (дерево папок), и отображает содержимое выделенной папки.

### *Структура окна в Windows.*

**Окно** - прямоугольная область экрана, в которой может отображаться приложение, документ или сообщение.

Окно может быть представлено в виде: свернутое на панель задач (минимизировано), нормальное (окно с обрамлением), полноэкранный экран (распахнутое на весь экран).

Любое окно Windows имеет ряд стандартных элементов.

*Системное меню* содержит команды для изменения размеров окна, его перемещения, минимизации, закрытия.

*Заголовок окна* является удобным управляющим элементом для перемещения окна. В заголовке окна отображается название приложения и документа (имя открытого файла).

*Строка меню* содержит пункты для открытия *ниспадающего меню*.

*Контекстное меню (всплывающее меню)* появляется при нажатии правой кнопки мыши и содержит активные команды для данного объекта.

*Рабочее поле* окна представляет собой область для размещения окна документа, и которая будет пустой, пока ни один документ не открыт.

*Рамка окна* - двойная линия, обрамляющая нормальное окно. Она служит управляющим элементом для изменения размеров окна.

*Линейки (полосы) прокрутки* окаймляют левую и нижнюю стороны окна и служат для перемещения (прокрутки) документа по вертикали и горизонтали.

*Панели инструментов* представляют собой линейки командных кнопок, предназначенных для быстрого вызова той или иной команды мышью.

*Строка состояния* находится у нижнего края окна и содержит информацию о режимах работы приложения.

**Активное окно** - окно приложения, реагирующее в данный момент на действия пользователя. Заголовок активного окна отличается по цвету и яркости от неактивного.

**Диалоговое окно** служит для ввода дополнительных параметров, необходимых для выполнения какой-либо команды.

**Окна приложений** представляют собой интерфейсы работающих приложений. Главным свойством окон приложений является то, что они могут перекрывать друг друга и являются независимыми, т.е. не подчинены никакому другому окну.

**Окна документов** всегда подчинены окнам своих приложений и не выходят за их пределы.

**Вспомогательные окна** используются для вывода сообщений.

### **Обмен данными между приложениями**

Передача фрагмента данных от одного документа в другой представляет собой простейший способ обмена данными между приложениями. Выполняется эта операция через буфер обмена.

**Буфер обмена (Clipboard)** - это специальным образом организованное динамическое пространство оперативной памяти для временного размещения данных и сведения, к какому программному приложению они относятся.

Для занесения информации в буфер обмена и ее извлечения оттуда имеются команды КОПИРОВАТЬ, ВЫРЕЗАТЬ, ВСТАВИТЬ.

### **Технология DDE - динамический обмен данными.**

Технология DDE опирается на связь, установленную между фрагментами одного документа, а также между документами различных приложений. Наличие такого механизма связи позволяет автоматически актуализировать данные в документе-приемнике при изменениях в источнике данных.

Например, в файле - источнике (графическом редакторе) открывают файл рисунка, фрагмент которого выделяется и копируется в буфер обмена. Предполагается вставка этого фрагмента в файл-приемник (текстовый документ) в качестве иллюстраций. При последующей команде ВСТАВКА/ ОБЪЕКТ/ИЗ ФАЙЛА фрагмент визуально отображается в текстовом документе, при этом создается ссылка на файл-источник. Все изменения рисунка автоматически отображаются и в документе-приемнике.

### **Технология OLE -связывание и встраивание объектов.**

Технология OLE опирается на связь, установленную между документом и приложением, а также на механизм, разработанный для связи приложений. Наличие такого механизма позволяет перенести в выходной документ (файл-приемник - OLE-клиент) фрагмент или файл другого приложения (файл-источник - OLE - клиент), при этом он приобретает свойства встроенного или связанного объекта. При этом при обращении к встроенным данным (например, щелчок мыши) происходит автоматический того приложения, к которому они относятся.

## ЛЕКЦИЯ №6

### Тема: Базы данных.

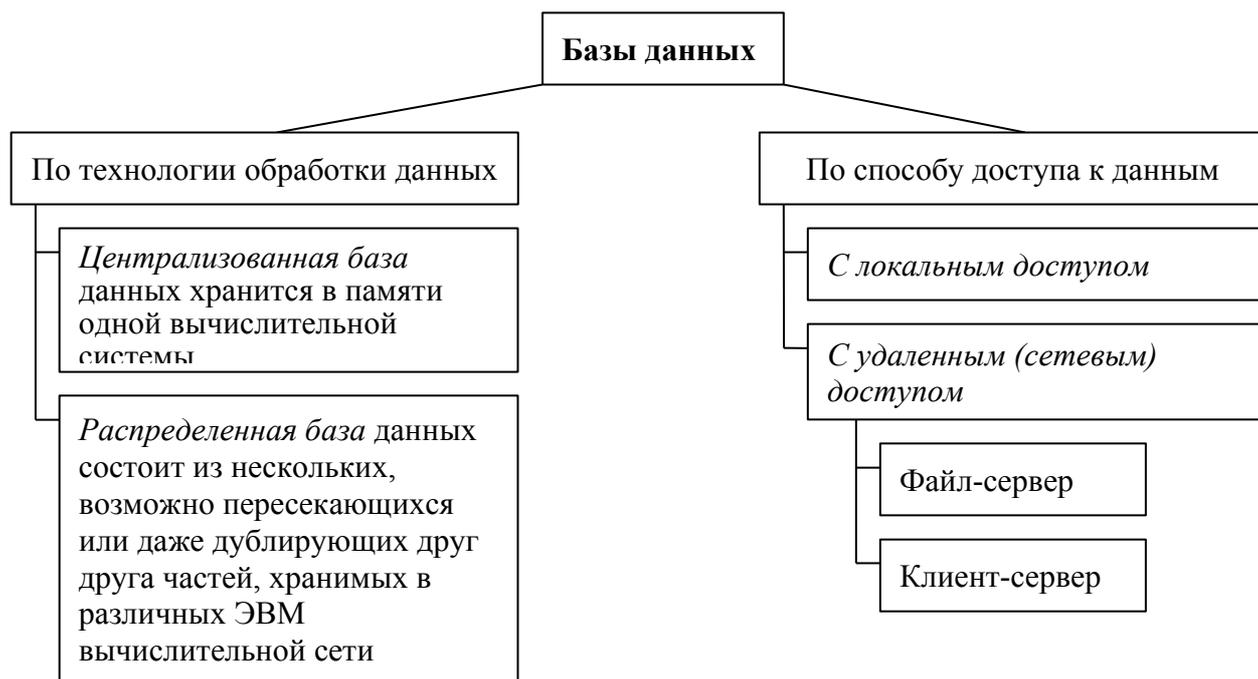
**База данных (БД)** – это поименованная совокупность структурированных данных, относящихся к определенной предметной области (БД по металлургии, БД в деканате о студентах, БД в библиотеке по книгам).

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

Основными функциями системы управления базами данных являются:

- создание структуры базы данных;
- предоставление средств для ее заполнения или импорта данных из другой базы;
- обеспечение возможности доступа к данным;
- предоставление средств поиска, фильтрации, запросов данных.

### Классификация БД:



### Структурные элементы базы данных:

**Поле** – элементарная единица логической организации данных, которая соответствует неделимой единице информации – реквизиту. Для описания поля используются следующие характеристики:

*Имя*, например, Фамилия, Имя, отчество, Дата рождения;

*Тип*, например, символьный, числовой, календарный;

*Длина*, например, 15 байт;

*Точность* для числовых данных, например для десятичного знака для отображения дробной части числа.

*Типы данных*, которые могут храниться в полях БД: числовые поля, поля даты и времени, текстовые поля, логические поля, поле текстовых примечаний.

**Запись** – совокупность логически связанных полей. Экземпляр записи - отдельная реализация записи, содержащая конкретные значения ее полей.

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

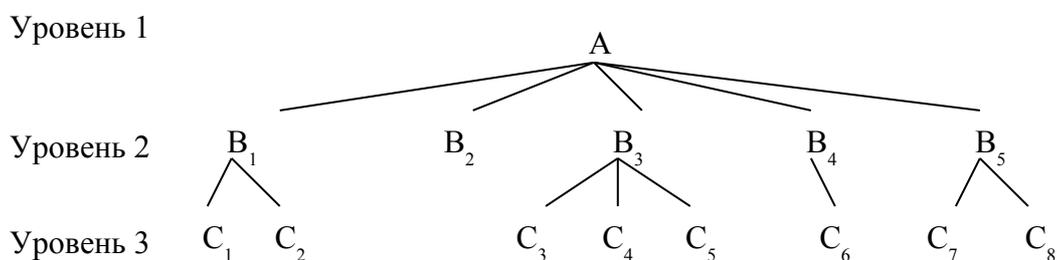
В структуре записи файла указываются поля, значения которых являются *ключами*: *первичными* (ПК), которые идентифицируют экземпляр записи, и *вторичными* (ВК), которые выполняют роль поисковых или группировочных признаков (по значению вторичного ключа можно найти несколько записей).

### Виды моделей данных.

**Модель данных** – совокупность структур данных и операций их обработки.

#### 1. Иерархическая модель данных.

Представляет собой совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево).



К основным понятиям иерархической структуры относятся: уровень, элемент(узел), связь.

*Узел* – это совокупность атрибутов данных, описывающих некоторый объект (на схеме – вершины графа). Каждый узел на более низком уровне может быть связан только с одним узлом, находящемся на более высоком уровне.

Приведем пример иерархической структуры БЗ.

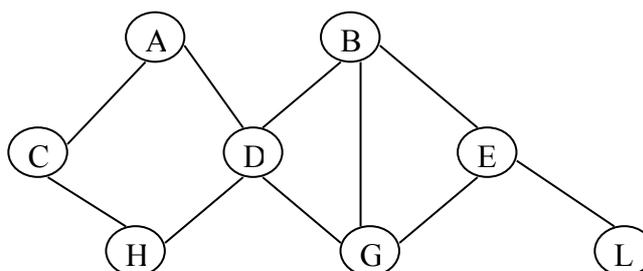
**Институт**(специальность, название директор)



Можно сказать, что каждый студент учится в определенной (только одной) группе, которая относится к определенному (только одному) институту.

2. *Сетевая модель данных.*

Объекты в сетевой модели связаны разнородно, т. е. при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.



Примером сетевой структуры БЗ может служить структура базы данных, содержащей сведения о студентах, участвующих в научно-исследовательских работах (НИРС). Возможно участие одного студента в нескольких НИРС, а также участие нескольких студентов в разработке одной НИРС (схему зарисовать самостоятельно!).

3. *Реляционная модель данных.*

Эта модель ориентированна на организацию данных в виде двумерных таблиц. Каждая *реляционная таблица* представляет собой двумерный массив и обладает следующими свойствами:

- ✓ каждый элемент таблицы – один элемент данных;

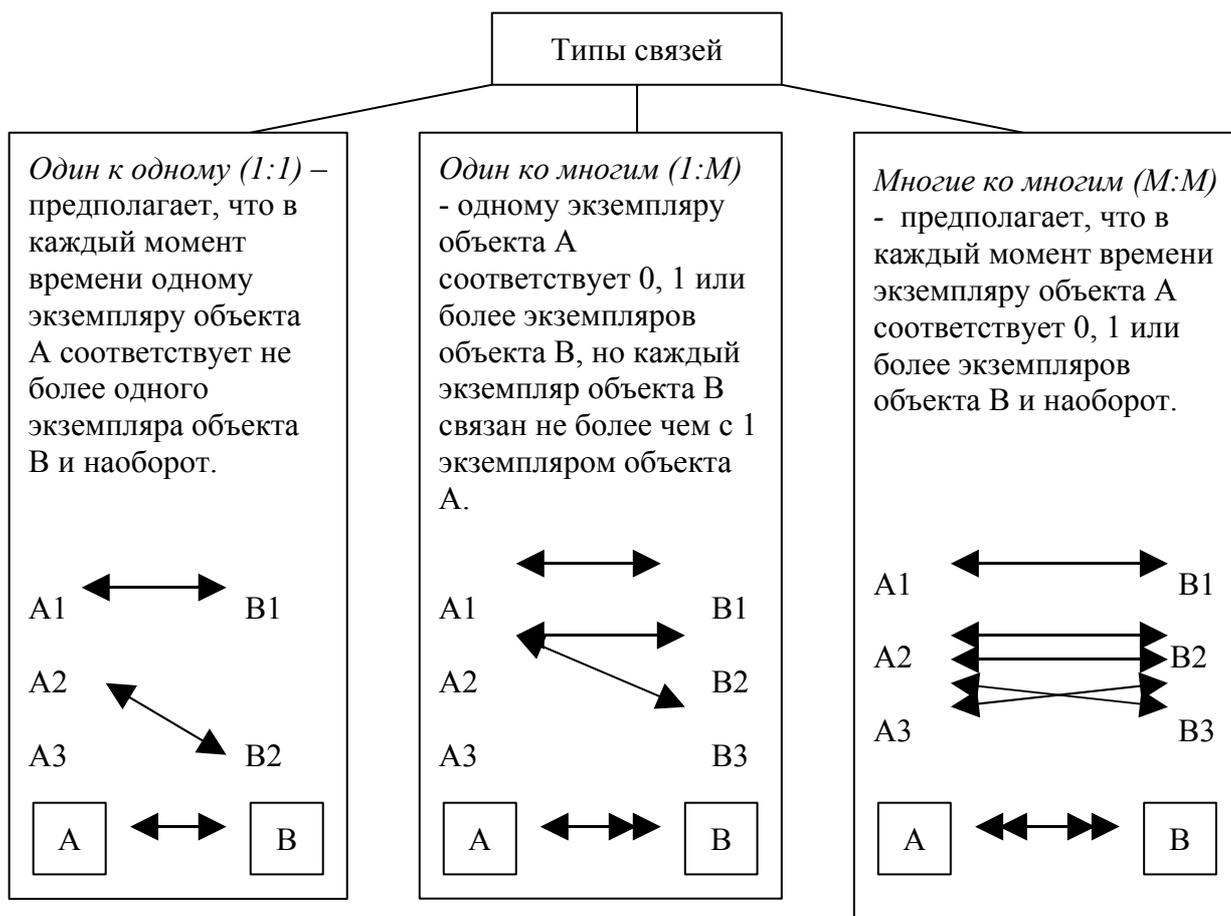
- ✓ все столбцы в таблице однородные, т. е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
- ✓ каждый столбец имеет уникальное имя;
- ✓ одинаковые строки в столбце отсутствуют;
- ✓ порядок следования строк и столбцов может быть произвольным.

Реляционной таблицей можно представить информацию о студентах, обучающихся в вузе.

Отношения представлены в виде *таблиц*, строки которых соответствуют кортежам или *записям*, а столбцы – атрибутам отношений, доменами, *полями*.

Поле, каждое значение которого однозначно определяет соответствующую запись, называется *простым ключом* (ключевым полем). Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет *составной ключ*. В примере ключевым полем таблицы является «№ личного дела».

Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей); в противном случае нужно ввести в структуру первой таблицы *внешний ключ* – ключ второй таблицы.



Гражданин - паспорт

Покупатель - торговый агент

Студент-преподаватель

К реляционной модели данных относится *Ms-Access*. В Access можно работать в следующих режимах:

- ✓ режим таблиц (можно использовать буфер обмена, фильтрацию данных, копирование полей и строк таблицы, перенос таблицы в другие базы данных);
- ✓ режим конструктора (определяет свойство объекта базы данных);
- ✓ режим запроса (простой, с параметром, перекрестный);
- ✓ режим отчета (используется для создания автоотчета, построения диаграмм);
- ✓ режим макроса (создание кнопки).

## ЛЕКЦИЯ №7

**Тема: алгоритмизация, языки программирования, компиляторы и интерпретаторы, поколения языков программирования.**

### Алгоритмизация

#### *Понятие алгоритма, его свойства, способы представления*

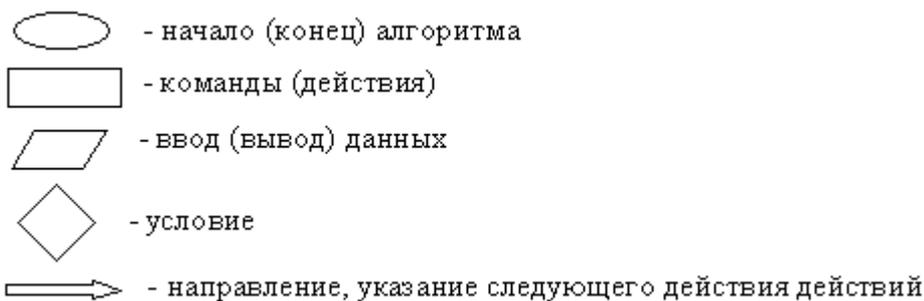
**Алгоритм** - понятное и точное предписание к исполнителю совершить последовательность действий, направленных на решение задач определенного типа.

#### **Свойства алгоритма:**

- *дискретность* (алгоритм должен быть разбит на конкретные действия, выполнив один шаг, приступает к исполнению следующего);
- *понятность* (точное понятие команды);
- *определенность* (исполнитель алгоритма не должен сомневаться в следующем шаге);
- *результативность* (по завершении выполнения алгоритма должен быть получен результат);
- *массивность* (алгоритм должен быть написан для решения ряда подобных задач).

**Способы представления алгоритма:** словесный, математическая формула, табличный, графический (блок-схема).

*Блок-схема представляет графический способ алгоритма.* Для записи блок-схемы используются следующие обозначения:



#### **Основные базовые структуры алгоритма**

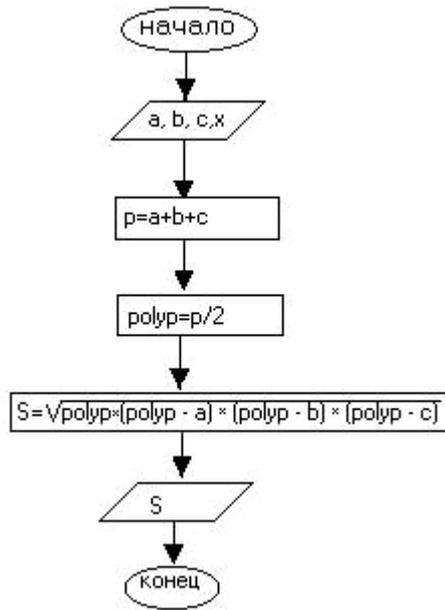
Основными базовыми структурами алгоритма являются:

- линейная структура;

- разветвляющая структура (ветвление);
- циклическая структура (цикл).

**Линейная структура** - алгоритмическая структура, в которой команды выполняются последовательно.

Примером линейной алгоритмической структуры может служить построение алгоритма нахождения площади треугольника по сторонам треугольника  $a, b, c$ . Площадь вычисляется с использованием формулы Герона.

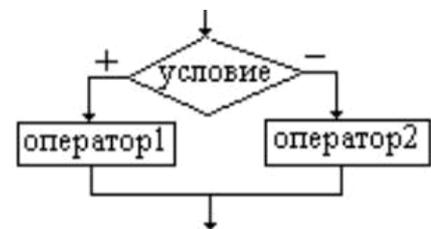


**Разветвляющая структура (ветвление)** - это алгоритмическая структура, в которой проверяется некоторое условие и в зависимости от результатов проверки выполняется то или иное действие.

Существует два типа ветвления: *полное* и *неполное*.

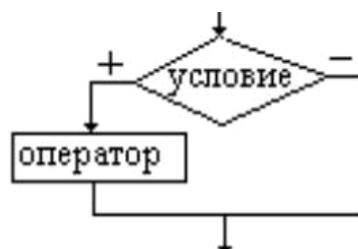
Алгоритм полного ветвления работает по следующему принципу. Вначале вычисляется условие. Если результат условия *True* (истина), то выполняется оператор1, а оператор2 пропускается. Если результат условия *False* (ложь), то выполняется оператор2, а оператор1 пропускается.

Блок-схема полного ветвления имеет следующий вид:



Блок-схема неполного ветвления имеет следующий вид:

Неполная структура работает по алгоритму. Сначала



разветвляющая следующему вычисляется условие.

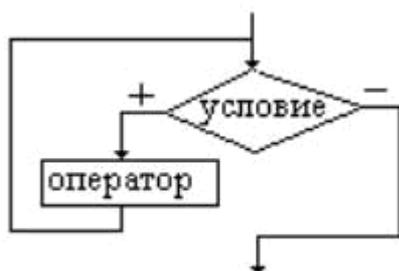
Если условие принимает значение TRUE, то выполняется оператор, иначе никаких действий не происходит.

**Циклическая структура (цикл)** – это алгоритмическая структура, содержащая повторяющиеся команды.

Циклические алгоритмические структуры бывают: цикл с предусловием, цикл с постусловием.

Во многих языках программирования используется циклическая структура с параметром (счетный цикл), который не является классической алгоритмической структурой.

Блок-схема цикла с предусловием следующая:



Циклическая структура с предусловием работает по следующему алгоритму. Первоначально проверяется условие. Если оно истинно, то выполняется тело цикла (оператор). Затем снова проверяется условие и т.д. Если условие ложно, то цикл завершается и выполняется оператор, стоящий непосредственно после цикла.

Блок-схема цикла с постусловием следующая:



Циклическая структура цикла с постусловием работает по следующему алгоритму: сначала выполняется тело цикла (операторы), потом проверяется условие. Если условие ложно, то снова выполняются операторы, и так до тех пор, пока условие не примет значение *TRUE*.

### **Моделирование решения задач. Этапы проектирования и создания программ**

**Формулируются и анализируются требования к проекту.** Этот этап самый важный, так как неправильная постановка цели проекта заставляет выполнять ненужную работу. На данном этапе определяются входные данные (их количество должно быть минимально), тип

этих данных, результирующие (выходные данные) и их приблизительная оценка. На основе требований по разным методикам определяется примерный объем проекта и его трудоемкость, рассчитываются будущие трудозатраты и определяется его стоимость (если речь идет о программном продукте).

**Выбирается методология разработки программы.** Используя метод структурного программирования, решаемую основную задачу разбивают на отдельные подзадачи, которые могут быть объединены в модули. При использовании RAD-средств проектируется интерфейс пользователя, определяются объекты управления программой, события и реакция на эти события. Выбираются события и определяются подзадачи, которые при этом событии будут решаться.

**Алгоритмизация.** Разработка алгоритмической структуры каждой подзадачи.

**Программирование.** Реализация конкретной подзадачи на языке программирования, возможно в виде реакции на событие.

**Тестирование и отладка.** Когда программа закончена (готова работоспособная *альфа-версия*), она тестируется. Тестирование – обнаружение ошибок. Исправляются синтаксические ошибки, ошибки программы. Задавая входные данные, полученный результат сравнивают с ожидаемым, в случае расхождения приступают к отладке. Анализируются, в частности, устойчивость работы программы при вводе недопустимых или критических значений, при отсутствии информации (не указании данных), при неверных действиях. Когда число ошибок становится минимальным, начинается *бета-тестирование*. К такому тестированию привлекаются максимально возможное число сотрудников, и программа уже начинает частично функционировать в рабочем режиме.

**Исполняемая программа.** Для программы необходимо подготовить сопровождение – справочную информацию о разработчике, назначение и правила работы с программой.

**Контроль качества.** На этапе тестирования и отладки нельзя дать стопроцентную гарантию качества программы. Чем крупнее проект, тем больше в нем ошибок. Предусматривается непрерывный, сквозной контроль качества за программой: вносятся изменения, конфигурируется новыми решаемыми подзадачами – выпускается новая обновленная версия. В ходе функционирования программы постоянно должны появляться новые версии (раз в 2-3 года) для поддержания спроса и удовлетворения потребностей на возникающие новые решаемые задачи в рамках программы.

### **Языки и системы программирования**

**Язык программирования** - формальный язык для описания алгоритма решения задачи на компьютере.

#### ***Уровни языков программирования***

Языки программирования бывают двух уровней: низкого и высокого уровня.

*Языки программирования низкого уровня* близки к машинному коду и ориентированы на конкретные команды процессора.

Языком самого низкого уровня является *язык ассемблера*, который просто представляет каждую команду машинного кода, но не в виде чисел, а с помощью символьных условных обозначений, называемых *мнемоникой*. Каждой модели процессора соответствует свой язык ассемблера.

С помощью языков низкого уровня создаются очень эффективные и компактные программы, так как разработчик получает доступ ко всем возможностям процессора. С другой стороны, для этого требуется очень хорошо понимать устройство компьютера (отладка больших приложений затруднена, а результирующая программа не может быть перенесена на компьютер с другим типом процессора). Подобные языки обычно применяются для написания небольших системных приложений, драйверов устройств, когда важнейшими требованиями становятся компактность, быстродействие и возможность прямого доступа к аппаратным ресурсам.

*Языки программирования высокого уровня* значительно ближе и понятнее человеку, для описания алгоритма используется привычная для него форма. Особенности конкретных компьютерных архитектур (аппаратных средств) в них не учитываются, поэтому создаваемые тексты программ легко переносимы на другие платформы, имеющие программу перевода данного языка в машинный код.

### ***Интерпретаторы и компиляторы***

С помощью языка программирования создается не готовая программа, а только ее текст, описывающий ранее разработанный алгоритм. Чтобы получить работающую программу, надо этот текст перевести в машинный код. Для этого используются программы-трансляторы, которые бывают двух видов: компиляторы и интерпретаторы.

*Программа-компилятор* переводит исходный текст программы в машинный код и записывает его на диске в виде отдельного бинарного файла (информация в машинном коде). При запуске программы исполняется бинарный файл. Почти все системы программирования работают с компилятором.

*Программа-интерпретатор* работает совместно с исходным текстом. Каждая команда интерпретируется в машинный код и немедленно исполняется. Файл на машинном языке не создается. Программа, написанная в системе программирования, включающая интерпретатор, работает медленно. Поэтому интерпретаторы не нашли широкого применения.

В настоящее время с интерпретатором работают в основном языки программирования для Интернета. Например, с интерпретатором работает Java, Perl.

### *Поколения языков программирования*

**1-е поколение** составляют языки, созданные в начале 50-х годов, когда первые компьютеры только появились на свет. Это был первый язык ассемблера, созданный по принципу «одна инструкция – одна строка».

**2-е поколение** составляют языки программирования конца 50-х – начала 60-х гг. Тогда появился символический ассемблер, в котором существовало понятие переменных. Он стал полноправным языком программирования.

**3-е поколение** языков программирования относится к 60-м гг. В это время родились универсальные языки высокого уровня, с помощью которых появилась возможность решать задачи из любых областей. Такие качества языков программирования высокого уровня как относительная простота, независимость от конкретного компьютера и возможность использования алгоритмических конструкций позволили резко повысить производительность труда программистов. Подавляющее большинство языков программирования этого поколения успешно применяется и сегодня.

**4-е поколение** языков программирования предназначено для реализации крупных проектов, повышена их надежность и скорость создания. 4-е поколение началось в 70-х гг. и продолжается до настоящего времени. Как правило, языки 4-го поколения имеют мощные функции (операторы, команды), для реализации которых на языках поколения младшего уровня потребовалось бы тысячи строк.

**5-е поколение** языков программирования появилось в середине 90-х гг. К ним относятся системы автоматического создания прикладных программ с помощью визуальных средств разработки, без знания программирования. Инструкции вводятся в компьютер в максимально наглядном виде с помощью методов, наиболее удобных для человека, не знакомого с программированием.

### *Обзор языков программирования высокого уровня*

**Fortran (Фортран)** – это компилируемый язык, созданный в 50-е гг. Этот язык появился первым после языка ассемблера, поэтому удобство создания программы не так хорошо реализовано, как возможность получения эффективного машинного кода. В 2000 г. выпущен Фортран F2k.

**Cobol (Кобол)** – компилируемый язык, разработанный в начале 60-х гг. для применения в экономической области и решения бизнес-задач. Отличается большой «многословностью» – его операторы иногда выглядят как обычные английские фразы. Очень распространен среди программистов США.

**Algol (Алгол)** – компилируемый язык, созданный в 1960 г. Он был призван заменить Фортран, но из-за более сложной структуры не получил широкого распространения.

**Pascal (Паскаль)** – создан в конце 70-х гг. Во многом напоминает Алгол, но в нем ужесточены требования к структуре программы, за счет чего структура стала более наглядной и простой. Паскаль удобен для получения азов программирования, но также успешно применяется при создании крупных проектов.

**Basic (Бейсик)** – создавался в 60-х гг. в качестве учебного языка и очень прост в изучении. Для него имеются интерпретаторы и компиляторы.

**C (Си)** – создан в лаборатории Bell и первоначально планировался для замены ассемблера. Имеет возможность создавать эффективные и компактные программы, в то же время не зависит от конкретного типа процессора. Си во многом похож на Паскаль и имеет дополнительные средства для прямой работы с памятью. В 1980 г. создано объектно-ориентированное расширение языка Си – C++(Си<sup>++</sup>).

**Java (Джава, Ява)** – язык, созданный компанией Sun в начале 90-х гг. на основе Си<sup>++</sup>. Главная особенность этого языка – компиляция не в машинный код, а в платформенно-независимый байт-код (каждая команда занимает 1 байт). Этот байт-код выполняется с помощью интерпретатора – визуальной Java-машины, версии которой существуют сегодня для любой платформы. Благодаря наличию множества Java-машин, программы легко переносятся на уровне двоичного байт-кода. Сегодня язык Ява очень популярен. Пока основной его недостаток – невысокое быстродействие, так как язык Ява интерпретируемый.

**Языки программирования баз данных** – язык структурированных запросов SQL. Основан на мощной математической теории и позволяет выполнять эффективную работу, манипулируя не отдельными, а группами записей. Для управления большими базами данных используются СУБД. Практически в каждой СУБД, помимо поддержки языка SQL, имеется свой уникальный язык, ориентированный на особенности этой СУБД: Microsoft, IBM, Oracle, Software.

**Языки программирования для Интернет** отличаются характерными особенностями: являются интерпретируемыми, интерпретаторы для них распространяются бесплатно, а сами программы в исходных текстах. Такие языки называются *скрипт-языками*. Примерами языков программирования для Интернет являются HTML, Perl, Tcl/Tk, VRML.

**Логические языки программирования** используются в области создания искусственного интеллекта. Наиболее распространены Пролог, Ада.

### ***Системы программирования***

Для написания программы на языке программирования необходимо иметь на компьютере установленную соответствующую систему программирования.

**Системы программирования** — хорошо интегрированная система, включающая как минимум:

специализированный текстовый редактор (для написания текста программы);

компилятор для перевода текста программы в машинный код (в редких случаях — интерпретатор);

библиотека функций (подключенные модули);

редактор связей для связывания модулей (файлов с исходными текстами) и стандартных функций, находящихся в библиотеках;

исполнимый код - законченную программу с расширением .COM или .EXE, которую можно запустить на любом компьютере, где установлена операционная система, для которой эта программа создавалась;

справочную систему;

отладчик, позволяющий анализировать работу программы во время ее выполнения по шагам.

В последние несколько лет в программировании (особенно в программировании для операционной системы Windows) наметился так называемый *визуальный подход*. Он облегчает создание графических приложений, в таких системах имеется множество стандартных элементов управления и контроль за их работой. Подобные системы программирования называются *средами быстрого проектирования RAD-среды*.

Наиболее популярны следующие визуальные среды быстрого проектирования программ для Windows:

Basic: Microsoft Visual Basic

Pascal: Borland Delphi

C++: Borland C++ Bulider

Java: Symantec Café.

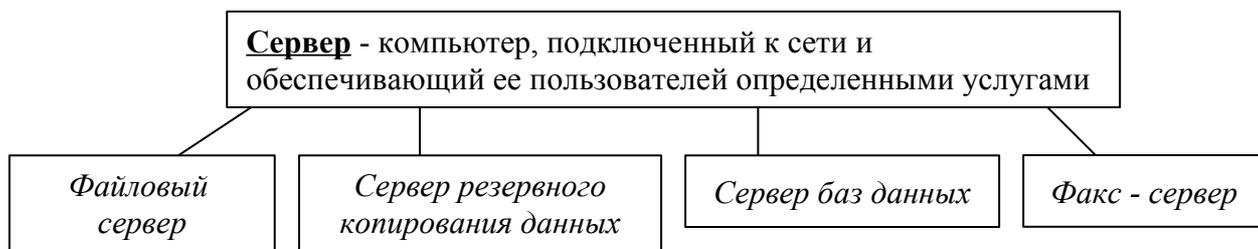
## ЛЕКЦИЯ №8

**Тема: Компьютерные сети.**

**Компьютерная (вычислительная) сеть** - совокупность компьютеров и терминалов, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных.

### **Устройства компьютерной сети**

Все устройства, подключаемые к сети, можно разделить на три функциональные группы: рабочие станции; серверы сети; коммуникационные узлы.

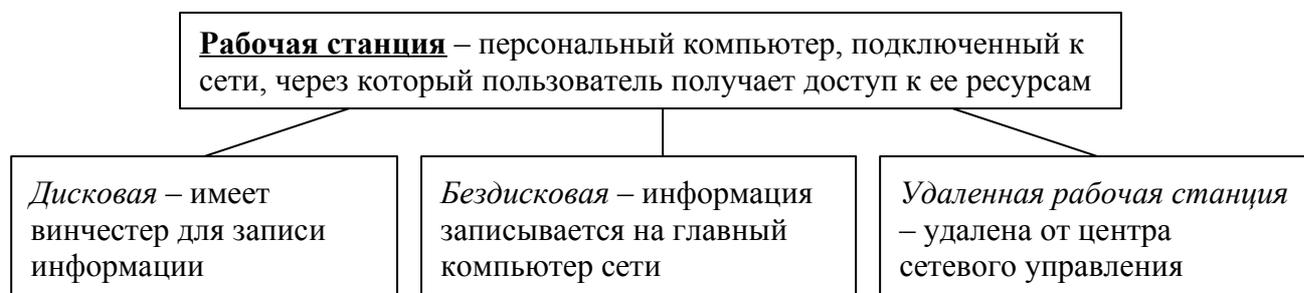


*Файловый сервер* – компьютер, хранящий данные пользователей сети и обеспечивающий доступ пользователей к этим данным. Как правило, компьютер, выделенный под файловый сервер, имеет большой объем дискового пространства. Файловый сервер выполняет следующие функции: хранение данных, архивирование данных, согласование изменений данных, выполняемых разными пользователями.

*Сервер баз данных* – компьютер, выполняющий функции хранения, обработки и управления файлами баз данных. Сервер баз данных выполняет следующие функции: хранение баз данных; прием и обработку запросов к базам данных, а так же пересылку результатов обработки на рабочую станцию; обеспечение авторизованного доступа к базам данных, поддержку системы ведения и учета пользователей, разграничение доступа пользователей; согласование изменений данных, выполняемых разными пользователями; поддержку распределенных баз данных, взаимодействие с другими серверами баз данных, расположенных в другом месте.

*Факс-сервер* – устройство или компьютер, который выполняет рассылку и прием факсимильных сообщений для пользователей локальной сети.

*Сервер резервного копирования данных* – устройство или компьютер, который решает задачи создания, хранения и восстановления копий данных с файлового сервера или рабочих станций.



*На рабочей станции с диском* (жестким или гибким) операционная система загружается с этого локального диска. Каждая дисковая рабочая станция работает со своими файлами на диске и использует свою операционную систему.

*Для бездисковой станции* операционная система загружается с диска файлового сервера. Без дисковая рабочая станция не имеет диска для хранения информации, поэтому обрабатывает файлы только с файлового сервера.

*Удаленная рабочая станция* – это станция, которая подключается к локальной сети через телекоммуникационные каналы связи (например, с помощью телефонной сети).

К **коммуникационным узлам** относятся следующие устройства: повторители, коммутаторы (мосты), маршрутизаторы, шлюзы.

Информация передается по сети порциями (пакетами). В сети при передаче отдельного пакета с помощью каналов связи происходит затухание сигнала. Поэтому сеть ограничивают участками (сегментами).

По окончании каждого участка имеется повторитель – устройство, усиливающее или регенерирующее пришедший на него сигнал. Повторитель, приняв пакет от одного сегмента, передает его дальше. Повторитель осуществляет связывание сегментов.

*Коммутатор* или *мост* – это устройство, которое, как и повторитель, выполняет объединение нескольких сегментов. В отличие от повторителя, мост выполняет развязку присоединенных к нему сегментов.

*Маршрутизатор* – устройство, соединяющее сети одного или разных типов по одному протоколу данных, анализирует адрес назначения и направляет данные по оптимально выбранному маршруту.

*Шлюз* – это устройство, позволяющее организовать обмен данными между разными сетевыми объектами, использующими разные протоколы обмена данными.

### **Основные показатели сети**

*Производительность* сети определяется количеством информации, переданной через сеть или её сегмент в единицу времени.

*Надежность* работы вычислительной сети определяется надежностью работы всех ее компонентов. Для обеспечения надежности, как правило, информация в сети храниться в нескольких экземплярах (дублирование информации). При передаче информации возможны потери и искажения. Для оценки надежности исполнения этой функции используются показатели вероятности потери пакета при его передаче, либо вероятность доставки пакета.

*Безопасность* – это способность сети обеспечить защиту информации от несанкционированного доступа. Задачи обеспечения безопасности решаются применением

специального программного обеспечения, использование более безопасной передающей среды в сети.

*Управлением* сетью занимается администратор сети, обычный пользователь не имеет административных прав. Администратор сети должен иметь возможность воздействовать на работу любого элемента сети. *Управляемость* сетью – возможность определения проблем в работе вычислительной сети или отдельных ее сегментов, выработка управленческих действий для решения выявленных проблем.

*Расширяемость* определяется возможностью добавления новых элементов сети (рабочих станций, служб).

*Масштабируемость* сети определяет возможность расширения сети без существенного снижения ее производительности. Например, одноранговые сети обладают хорошей расширяемостью, но плохой масштабируемостью.

*Интегрируемость* означает возможность подключения к вычислительной сети разнообразного и разнотипного оборудования.

#### Классификация вычислительных сетей:



#### Типы передающей среды в компьютерных сетях

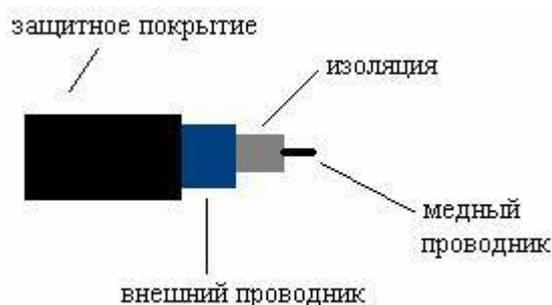
В компьютерных сетях используется два основных вида передающей среды: кабельный канал связи и радиосвязь.

## Кабельный вид связи

*Кабельный вид* связи используют преимущественно в локальных, чаще региональных сетях. Кабели бывают нескольких видов: витая пара проводов, коаксиальный кабель, оптоволоконный кабель.

*Витая пара* состоит из двух изолированных проводов, свитых между собой. Такой вид соединения недорогой. Недостаток витой пары - плохая помехозащищенность и низкая скорость передачи информации. Существует неэкранированная витая пара и экранированная витая пара (рис.9.2). Дополнительный защитный слой в экранированной паре делает кабель помехоустойчивым, значительно уменьшает электромагнитное излучение. Кабели витой пары классифицируются по производительности передачи данных. В вычислительных сетях используют кабели витой пары уровня 3 – производительностью до 16 Мбит/с, уровня 4 – до 20 Мбит/с, уровня 5 – до 100 Мбит/с.

**Коаксиальный кабель** по сравнению с витой парой обладает большей прочностью, помехозащищенностью. В центре коаксиального кабеля находится медный проводник, окруженный толстым слоем изоляционного материала. Второй слой сделан в виде оплетки поверх изоляции.



Коаксиальные кабели выпускают разных типов. Различают «толстый» и «тонкий» коаксиальный кабель, в зависимости от диаметра сердечника. «Толстый» кабель обладает лучшими характеристиками, чем «тонкий». Скорость передачи данных по коаксиальному кабелю до 10-50 Мбит/с.

**Оптоволоконный кабель** не подвержен действию электромагнитных полей. В оптоволоконном кабеле для передачи данных используются световые импульсы. Сердечник такого кабеля изготовлен из стекла или пластика (рис.9.4). Сердечник окружен слоем отражателя, который направляет световые импульсы вдоль кабеля путем отражения от стенок. Скорость передачи информации по оптоволоконному кабелю до 10 Гбит/с. Оптоволоконный кабель более дорогой по сравнению с предыдущими типами.



Различают одномодовый и многомодовый оптоволоконные кабели. В одномодовом кабеле более тонкий центральный сердечник, он пропускает один световой луч, который почти не отражается от внешнего отражателя. В многомодовом кабеле используется более толстый центральный сердечник, в котором одновременно существует несколько световых лучей с разными углами преломления.

### Радиосвязь

Каналы радиосвязи используют различные диапазоны передачи данных. В радиосвязи передача информации осуществляется от одного ретранслятора к другому. Разновидностью радиосвязи является спутниковая связь, в которой передача данных осуществляется через спутник.

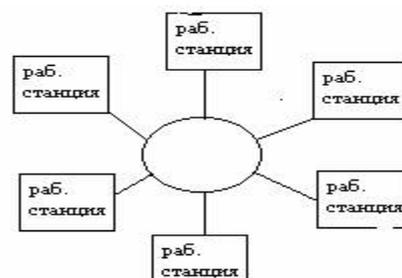
### Конфигурация локальных вычислительных сетей

Существуют следующие конфигурации ЛВС: шинная, кольцевая, звездообразная, древовидная. От конфигурации ЛВС зависит, как размещаются абоненты сети и как они соединяются между собой.

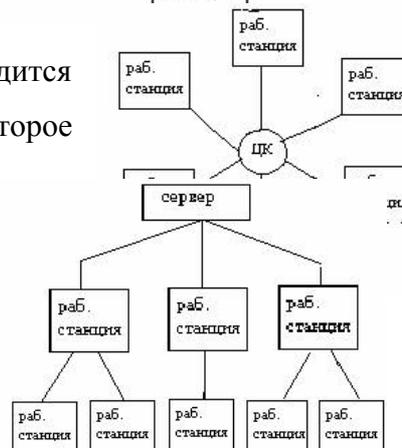
*Шинной* называется такая конфигурация сети, при которой к незамкнутому каналу (шине) с некоторым интервалом подключаются рабочие станции. Информация от абонента-источника распространяется по каналу в обе стороны.



В *кольцевой* ЛВС информация передается по замкнутому каналу (кольцу), в большинстве случаев только в одном направлении. Каждый абонент непосредственно связан с двумя соседними абонентами, но «прослушивает» передачу любого абонента сети.



*Звездообразные* ЛВС возникли на основе телефонных сетей с АТС. В центре звездообразной ЛВС находится центральный коммутатор, либо устройство, которое последовательно опрашивает абонентов и предоставляет им право на обмен информацией.



*Древовидную конфигурацию ЛВС* образуется путем подсоединения нескольких простых шин к одной магистральной при помощи мостов или шлюзов.

### **Системное программное обеспечение вычислительных сетей**

Для эффективной работы сетей используются специальные операционные системы, которые в отличие от персональных операционных систем, предназначены для решения специальных задач по управлению работой сети компьютеров. Это сетевые операционные системы, которые устанавливаются на специально выделенные компьютеры. Признанными лидерами сетевых ОС являются Windows NT, NowellNetWare, Unix.

Обработка информации в сети распределена между двумя объектами: клиент (рабочая станция) и сервер. Клиент может запросить с сервера чтение – запись файла, поиск информации и т.д. Сервер выполняет запрос, поступивший от клиента. Подобные системы принято называть системами "клиент – сервер" или **архитектура "клиент-сервер"**.

В зависимости от распределения функций между компьютерами в сети можно выделить одноранговые сетевые ОС и ОС с выделенным сервером.

Одноранговые ОС используются в **одноранговых сетях**, для которых характерно: нет единого центра управления и единого устройства для хранения данных, сетевая операционная система распределена по всем рабочим станциям. Каждая станция может исполнять роль клиента или сервера. *Достоинства*: низкая стоимость и высокая надежность. *Недостатки*: сложность управления сетью, сложность обеспечения защиты информации, трудности обновления программного обеспечения, ограниченность количества рабочих станций. Одноранговая сеть преимущественно используется в НИИ, фирмах, коллективах, в которых ведется совместная работа по изучаемому вопросу.

**Сеть с выделенным сервером.** В такой сети один компьютер выполняет роль сервера. На нем устанавливается сетевая операционная система, к нему подключаются все внешние устройства. Взаимодействие между рабочими станциями в сети осуществляется через сервер. Количество серверов может быть более одного. *Достоинства*: надежная система защиты информации, высокое быстродействие, отсутствие ограничений на число рабочих станций. *Недостатки*: высокая стоимость сервера, меньшая гибкость по сравнению с одноранговой сетью.

Любая коммуникационная сеть должна включать следующие основные компоненты: передатчик, сообщение, средства передачи, приемник.

**Передатчик** – устройство, являющееся источником данных.

**Приемник** – устройство, принимающее данные.

**Сообщение** – цифровые данные определенного формата, предназначенные для передачи.

**Средства передачи** – физическая передающая среда и специальная аппаратура, обеспечивающая передачу сообщений.

Для передачи сообщений в вычислительных сетях используются различные типы каналов связи. Наиболее распространены выделенные телефонные каналы и специальные каналы для передачи цифровой информации. Применяются также радиоканалы и каналы спутниковой связи.

Основное назначение любой компьютерной сети – предоставление информационных и вычислительных ресурсов подключенным к ней пользователям.

### **Глобальная сеть Интернет**

**Internet** представляет собой глобальную компьютерную сеть. Само ее название означает "между сетей". Это сеть, соединяющая отдельные сети.

Internet обеспечивает обмен информацией между всеми компьютерами, которые входят в сети, подключенные к ней. Тип компьютера и используемая им операционная система значения не имеют.

Для отправления сообщения адресату в нужную сеть используется устройство *маршрутизатор*.

### **Система адресации в Internet**

Internet самостоятельно осуществляет передачу данных. К *адресам станций* предъявляются специальные требования. Адрес должен иметь формат, позволяющий вести его обработку автоматически, и должен нести некоторую информацию о своем владельце.

С этой целью для каждого компьютера устанавливаются два адреса: цифровой IP-адрес (IP - Internetnetwork Protocol - межсетевой протокол) и доменный адрес.

Оба эти адреса могут применяться равноценно. Цифровой адрес удобен для обработки на компьютере, а домашний адрес - для восприятия пользователем.

*Цифровой адрес* имеет длину 32 бита. Для удобства он разделяется на четыре блока по 8 бит, которые можно записать в десятичном виде. Адрес содержит полную информацию, необходимую для идентификации компьютера.

Для блока определяют адрес сети, а два другие - адрес компьютера внутри этой сети. Существует определенное правило для установления границы между этими адресами. Поэтому IP - адрес включает в себя три компонента: адрес сети, адрес подсети, адрес компьютера в подсети.

*Доменный* адрес определяет область, определяющую ряд хост - компьютеров. В отличие от цифрового адреса он читается в обратном порядке. Вначале идет имя компьютера, затем имя сети, в которой он находится.

В системе адресов Internet приняты домены, представленные географическими регионами. Они имеют имя, состоящее из двух букв.

Пример Географические домены некоторых стран: Франция - fr; Канада - ca; США - us; Россия - ru.

Существуют и домены, разделенные по тематическим признакам. Такие домены имеют трехбуквенное сокращенное название.

Пример Учебные заведения - edu. Правительственные учреждения - gov. Коммерческие организации - com.

### **Службы Internet**

1. Электронная почта (E-Mail).
2. Списки рассылки (MailList).
3. Телеконференции (Usenet).
4. Всемирная паутина WWW.
5. Служба передачи файлов (FTP).
6. Chat (разговор) в реальном времени (IRC).
7. ICQ.

### **Электронная почта**

Эта служба является старейшей. Её обеспечением в Internet занимаются специальные почтовые серверы. Почтовые серверы получают сообщения от клиентов и пересылают их по цепочке к почтовым серверам адресатов, где эти сообщения накапливаются. Сообщения могут содержать не только текстовую информацию, но и присоединенные файлы других типов – документы, таблицы, звуковые, графические файлы объемом до 2 Мбайт. При установлении соединения между адресатом и его почтовым сервером происходит автоматическая передача поступивших сообщений на компьютер адресата. Наиболее популярны Microsoft Outlook Express, The Bat!, Eudora, Pegasus mail.

Формат адреса электронной почты должен иметь вид:

Имя [пользователя@адрес](#) хост-компьютера

### **Списки рассылки**

Обычная электронная почта предполагает наличие двух партнеров по переписке. Если партнеров нет, то большой поток почтовой информации можно получить в свой адрес, подписавшись на списки рассылки. Это специальные тематические серверы, собирающие информацию по определенным темам и переправляющие её подписчикам в виде сообщений

электронной почты. Темы списков рассылки могут быть различными: коммерческие финансовые обзоры, презентации новых программ и аппаратных средств и т.д.

### **Новости UseNet. Телеконференции**

**Телеконференции** (группы новостей) – это дискуссионные группы, члены которых делятся друг с другом своими знаниями, впечатлениями, проблемами.

Вся система телеконференций организована по иерархическому принципу и разбита на тематические группы. В Internet предлагается свыше 50.000 тем для обсуждения. Например, темами основных групп являются:

- comp – темы, связанные с компьютерами;
- sci – темы из области научных исследований;
- news – информация и новости Usenet;
- soc – социальная тематика;
- talk – дискуссии.

### **Всемирная паутина WWW – World Wide Web**

WWW – это единое информационное пространство, состоящее из взаимосвязанных электронных документов, хранящихся на Web-серверах. Отдельные документы, составляющие пространство Web, называют Web-страницами. Группы тематически объединенных Web-страниц называют Web-узлами.

Всемирная паутина появилась на свет в 1992 г. Была создана Тимом берсом-Ли из Европейского центра ядерных исследований, расположенного в Женеве, Швейцария.

Система связанных между собой страниц, представляющих Web – узлы всего мира. Программы для просмотра Web-страниц броузерами (обозреватели). Наиболее часто используемые броузеры Netscape Navigator, Microsoft Internet Explorer.

Основной особенностью WWW является использование гипертекста и возможность клиентов взаимодействовать с другими приложениями Internet.

*Гипертекст* – текст, содержащий в себе связи с другими текстами, графической, видео- и звуковой информацией. Связь между гипертекстовыми документами осуществляется с помощью ключевых слов. Найдя ключевое слово, пользователь может перейти в другой документ, который в свою очередь тоже будет иметь гипертекстовые ссылки.

Гипертекстовая связь осуществляется на основе того, что каждый документ в этом пространстве обладает своим уникальным адресом (URL-адресом). URL-адрес состоит из трех частей:

- 1 указание службы, которая осуществляет доступ к данному ресурсу;

2. указание доменного имени компьютера (сервера), на котором хранится данный ресурс;
3. указания полного пути доступа к файлу на данном компьютере.

Например: <http://seach.netscape.com/assist/security/fags/cookies.html>

Все серверы WWW используют специальный язык HTML (Hypertext Markup Language – язык разметки гипертекста).

Собственные Web-страницы можно создавать как с помощью обычного текстового редактора (например, блокнот), так и с помощью html-редакторов. Наиболее популярен html-редактор Front Page Express, запустить который можно командой Пуск/программы/Internet Explorer/Front Page Express.

### **Служба передачи файлов (FTP)**

Прием и передача файлов составляют значительный процент от прочих Internet-услуг. Необходимость в передаче файлов возникает, например, при приеме файлов программ, при пересылке крупных документов, а также при передаче архивных файлов.

Служба *FTP* имеет свои серверы в мировой сети, на которых хранятся архивы данных. Со стороны клиента для работы с серверами *FTP* может быть установлено специальное программное обеспечение, хотя в большинстве случаев браузеры WWW обладают встроенными возможностями для работы и по протоколу *FTP*.

Протокол *FTP* работает одновременно с двумя *TCP*-соединениями между сервером и клиентом. По одному соединению идет передача данных, а второе соединение используется как управляющее. Протокол *FTP* также предоставляет серверу средства для идентификации обратившегося клиента. Однако существуют и десятки тысяч *FTP*-серверов с анонимным доступом для всех желающих.

### **Разговор в реальном времени. Chat– беседы**

Служба IRC (Internet Relay Chat) предназначена для прямого общения нескольких человек в реальном режиме времени. Иногда службу IRC называют чат-конференциями или просто чатом. В отличие от телеконференций, в системе IRC общение происходит только в пределах одного канала, в работе которого принимают участие обычно лишь несколько человек. Каждый пользователь может создать собственный канал и пригласить в него участников «беседы» или присоединиться к одному из открытых в данный момент каналов. Существует несколько популярных клиентских программ для работы с серверами и сетями, поддерживающими сервис IRC. Одна из наиболее популярных - программа mICR.exe.

### **ICQ**

Эта служба предназначена для поиска сетевого IP-адреса человека, подключенного в данный момент к сети Internet. Название службы является акронимом выражения *I seek you* –

я тебя ищу. Для пользования этой службой надо зарегистрироваться на её центральном сервере (<http://www.icq.com>) и получить персональный идентификационный номер *UIN*. Если номер известен партнерам по контактам, то служба *ICQ* приобретает характер *Internet-пейджера*. Зная номер *UIN*, но не зная его текущий *IP-адрес*, можно через центральный сервер службы отправить ему сообщение с предложением установить соединение.

### **Проблемы безопасности информации**

**Защита информации** – контрольные механизмы, препятствующие незаконному использованию ресурсов.

Проведение финансовых операций с использованием Интернета или других сетей, заказ товаров и услуг, использование кредитных карточек, доступа к закрытым информационным ресурсам, передача телефонных разговоров требуют обеспечения соответствующего уровня безопасности. Начиная с 90-х годов, Интернет становится мощным средством обеспечения розничного торгового оборота, в нем циркулируют договорные и финансовые обязательства, необходимость защиты которых как от просмотра, так и от фальсификации, очевидна.

Конфиденциальная информация, которая передается по сети Интернет, проходит через определенное количество маршрутизаторов и серверов, прежде чем достигнуть пункта назначения. Существует возможность перехвата информации, более того, информация может быть изменена и перехвачена адресату в измененном виде.

Проблемы, возникающие с безопасностью передачи информации при работе в компьютерных сетях, можно разделить на следующие типы:

опасность заражения компьютерным вирусом;

перехват информации - целостность информации сохраняется, но ее конфиденциальность нарушена;

модификация информации - исходное сообщение изменяется либо полностью подменяется другим и отсылается адресату;

подмена авторства информации. Например, кто-то может послать письмо от вашего имени или Web-сервер может притворяться электронным магазином, принять заказы, номера кредитных карт, но не высылать никаких товаров.

Под термином "безопасность" подразумевается совокупность трех различных характеристик:

1. *аутентификация* - процесс распознавания пользователя системы и представления ему определенных прав и полномочий;

2. *целостность* - идентичность отправленного и принятого;

3. *секретность* - предотвращение несанкционированного доступа к информации.

## Методы защиты информации

### 1. Антивирусная защита

Защиту от вирусов можно разделить на два класса: общие и профилактические средства. К общим средствам относятся - создание копий файлов и системных областей дисков и разграничение доступа - предотвращает использование компьютера без разрешения. Профилактические средства – наиболее обширный класс, к которому относятся антивирусные программы. Антивирусные программы разделяются на мониторы и сканеры. Монитор - находится резидентно (постоянно) в оперативной памяти компьютера и сообщает об обнаруженном вирусе. Сканер - программа, запускаемая пользователем (когда следует проверить дисковые накопители) или автоматически (периодическая проверка дисков). Важно, чтобы антивирусные программы, используемые для проверки, были самых последних версий. Широко используемые антивирусные программы - Drweb, AVP, Norton Antivir и др.

### 2. Криптография

**Криптография** - *шифрование*, позволяющая трансформировать данные в зашифрованную форму, из которой извлечь исходную информацию можно только при помощи ключа. Обеспечивает секретность информации.

В основе шифрования лежат два основных понятия: алгоритм и ключ. *Алгоритм* - это способ закодировать исходный текст, в результате чего получается зашифрованное послание. Зашифрованное послание можно интерпретировать только с помощью ключа.

Существуют две основные схемы шифрования: *симметричное шифрование* и *несимметричное шифрование*.

При **симметричном шифровании** отправитель и получатель владеют одним и тем же ключом, с помощью которого они могут зашифровать и расшифровать данные. Основным недостатком симметричного процесса заключается в том, что прежде чем начать обмен информацией, надо выполнить передачу ключа, а для этого опять-таки нужна защищенная связь, то есть проблема повторяется, хотя и на другом уровне. Если рассмотреть оплату клиентом товара или услуги с помощью кредитной карты, то получается, что торговая фирма должна создать по одному ключу для каждого своего клиента и каким-то образом передать им эти ключи. Это крайне неудобно.

Поэтому в настоящее время в Интернете **используют несимметричное шифрование**. В данной схеме для шифрования послания используется один ключ (*открытый*), для расшифровки - другой (*закрытый, личный*). На самом деле это как бы две "половинки" одного целого ключа, связанные друг с другом.

Ключи устроены так, что сообщение, зашифрованное одной половинкой, можно расшифровать только другой половинкой (не той, которой оно было закодировано). Создав пару ключей, торговая компания широко распространяет *публичный* ключ (открытую половинку) и надежно сохраняет *закрытый* ключ (свою половинку).

Как публичный, так и закрытый ключ представляют собой некую кодовую последовательность. Публичный ключ компании может быть опубликован на ее сервере, откуда каждый желающий может его получить. Если клиент хочет сделать фирме заказ, он возьмет ее публичный ключ и с его помощью закодирует свое сообщение о заказе и данные о своей кредитной карте. После кодирования это сообщение может прочесть только владелец закрытого ключа. Никто из участников цепочки, по которой передается информация, не в состоянии это сделать. Даже сам отправитель не может прочитать собственное послание, хотя ему хорошо известно содержание. Лишь получатель сможет прочесть сообщение, поскольку только у него есть закрытый ключ, дополняющий использованный публичный ключ.

Если фирме надо будет отправить клиенту квитанцию о том, что заказ принят к исполнению, она закодирует ее своим закрытым ключом. Клиент сможет прочесть квитанцию, воспользовавшись имеющимся у него публичным ключом данной фирмы. Он может быть уверен, что квитанцию ему отправила именно эта фирма, и никто другой, поскольку никто иной доступа к закрытому ключу фирмы не имеет.

Число бит в ключе определяет надежность защиты. Например, 8-битный ключ допускает  $2^8=256$  комбинаций ключей. Если использовать 128-битный ключ, то необходимо будет перебрать  $2^{128}$  ключей, что в настоящее время не под силу даже самым мощным компьютерам.

Даже если послание зашифровано, все равно остается возможность модификации исходного сообщения или подмены этого сообщения другим. Защита публичным ключом не является абсолютно надежной. Дело в том, что поскольку каждый желающий может получить и использовать чей-то публичный ключ, то он может сколь угодно подробно изучить алгоритм работы механизма шифрования и попытаться установить метод расшифровки сообщения, то есть реконструировать закрытый ключ.

Во многих странах вопрос применения алгоритмов шифровки данных находится в поле законодательного регулирования. В частности, в России к использованию в государственных и коммерческих организациях разрешены только те программные средства шифрования данных, которые прошли государственную сертификацию в административных органах, в частности, в Федеральном агентстве правительственной связи и информации при Президенте Российской Федерации.

### 3. Электронная подпись

Мы рассмотрели, как клиент может переслать организации свои конфиденциальные данные. Точно также он может общаться и с банком, отдавая ему распоряжения о перечислении своих средств на счета других лиц и организаций. Однако здесь возникает проблема: как банк узнает, что распоряжение поступило именно от данного лица, а не от злоумышленника, выдающего себя за него? Одним из путей решения этой проблемы является передача получателю уникальной записи - *электронной подписи*.

Принцип ее создания тот же, что и рассмотренный выше. С помощью специальной программы (полученной от банка) создаются те же два ключа: *закрытый и публичный*. Публичный ключ передается банку. Если теперь надо отправить поручение банку на операцию с расчетным счетом, оно кодируется *публичным ключом* банка, а своя подпись под ним кодируется собственным *закрытым ключом*. Банк поступает наоборот. Он читает поручение с помощью своего закрытого ключа, а подпись - с помощью публичного ключа поручителя. Если подпись читаема, банк может быть уверен, что поручение ему отправили именно мы, и никто другой.

### 4. Сертификация Web-узлов

При заказе товаров в Интернете важно убедиться, что сервер, принимающий заказы и платежи от имени некоей фирмы, действительно представляет эту фирму. Подтвердить это может сертификат продавца. В сертификате указано, когда он выдан и на какой срок. Прежде чем выполнять платежи через Интернет или опрашивать данные о своей кредитной карте кому-либо, следует проверить наличие действующего сертификата у получателя путем обращения в сертификационный отдел. Это называется *сертификацией Web-узла*. Например, в обозревателе Microsoft Internet Explorer 5.0, доступ к центрам сертификации осуществляется командой *Сервис* → *Свойства Обозревателя* → *Содержание* → *Сертификатов* → *Доверенные Корневые Центры Сертификации*.

### 5. Аутентификация

Аутентификация является одним из самых важных компонентов организации защиты информации в сети. Прежде чем пользователю будет предоставлено право получить тот или иной ресурс, необходимо убедиться, что он действительно тот, за кого себя выдает.

При получении запроса на использование ресурса от имени какого-либо пользователя сервер, представляющий данный ресурс, передает управление серверу аутентификации. После получения положительного ответа сервера аутентификации пользователю предоставляется запрашиваемый ресурс.

Одной из схем аутентификации является использование *стандартных паролей*. Эта схема является наиболее уязвимой с точки зрения безопасности - пароль может быть

перехвачен и использован другим лицом. Чаще всего используются схемы с применением *одноразовых паролей*. Даже будучи перехваченным, этот пароль будет бесполезен при следующей регистрации, а получить следующий пароль из предыдущего является крайне трудной задачей.

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

Задачей преподавателя при проведении лабораторных работ является грамотное и доступное разъяснение принципов и правил проведения работ, побуждение студентов к самостоятельной работе, определения места информатики в дальнейшей профессиональной работе будущего специалиста.

Цель лабораторной работы – научить студентов самостоятельно производить необходимые действия для достижения необходимого результата.

Прежде чем приступить к выполнению лабораторной работы, студенту необходимо ознакомиться с теоретическим материалом, соответствующим данной теме.

Выполнение лабораторной работы целесообразно разделить на несколько этапов:

- Формулировка и обоснование цели работы;
- Определение теоретического аппарата, применительного к данной теме;
- Выполнение заданий;
- Анализ результата;
- Выводы.

#### **V. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ДОМАШНИХ ЗАДАНИЙ**

Внеаудиторная работа по информатике включает в себя:

- Совершенствование и закрепление теоретических знаний, полученных на лекционных и лабораторных занятиях. Каждая тема курса включает вопросы входного контроля знаний (минимальный теоретический уровень), освоение которых необходимо для решения учебных задач, формирования умений и навыков темы.
- Формирование навыков практической работы - доведение умений до автоматизма путем решения упражнений - заданий, требующее повторного выполнения действий с целью его усвоения.
- Выполнение творческих работ, предусмотренных рабочей программой (см. пункт самостоятельная работа студентов).

При выполнении домашней работы студенты могут использовать различные источники приобретения информации: конспекты лекций, учебно-методические материалы курса, ссылки на научную литературу в информационном пространстве Интернета и др.

#### **VI. ПЕРЕЧЕНЬ ПРОГРАММНЫХ ПРОДУКТОВ, ИСПОЛЬЗУЕМЫХ ПРИ ПРЕПОДАВАНИИ ДИСЦИПЛИНЫ**

Наименование	Год выпуска, разработчик	Примечание
Total Commander	2004	Файловый менеджер (для windows)
Microsoft Office	2000, Microsoft	Пакет прикладных программ
Internet Explorer	2001, Microsoft	Обозреватель Internet
MathCad	2001, MathSoft, inc	Математический пакет
WinZip	2004	Архиватор
WinRAR 3.2	2004	Архиватор
Delphi for Object Pascal.	2000, Borland Delphi	Система объектно-ориентированного программирования
CorelDraw	2003, Corel inc	Графический пакет

## VII. КОМПЛЕКТ ЗАДАНИЙ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ

### Windows Программа Поиск

#### Задания:

1. Найдите файлы по имени и типу.

- ????.exe
- D\*.xls
- Все файлы с именем не более чем из трех символов.
- Все файлы с расширением sys.
- Все файлы с именем, начинающимся с ab и состоящим не более чем из пяти символов.
- Все файлы, начинающиеся с символа A, имеющие в расширении три символа, последний символ t.
- Все файлы созданные в программе Paint.
- Все файлы, начинающиеся с буквы O.
- Графический файл, начинающийся на букву T.

Сколько найдено файлов в каждом из случаев, какого они типа, в каких программах созданы?

2. Найдите файлы созданные сегодня, вчера, за последнюю неделю.

3. Отчитайтесь по проделанной работе преподавателю.

### Программа Проводник. Панель управления.

#### Задания:

1. Запустите программу Проводник. Какие кнопки находятся на панели инструментов Проводника? Все последующие пункты работы выполняйте в Проводнике.

2. Измените Вид правой панели с помощью меню, а затем Панели инструментов на:

- мелкие значки;
- крупные значки;
- таблицу;
- список.

3. Пользуясь панелью инструментов, откройте все папки, содержащие вложенные папки; переходя по уровням, проследите, как меняется содержимое правой панели.

4. Прочитайте свойства некоторых папок.

5. Создайте папку с именем ПАПКА2.

6. Переименуйте ПАПКА2 в ПАПКА РАБОЧАЯ.

7. Скопируйте ПАПКА РАБОЧАЯ на диск C:.

8. Отсортируйте файлы по имени, размеру, дате.
9. Укажите курсором файл, имеющий самый маленький размер в своей папке, папке диска С:.
10. Очистите список документов в Главном меню.
11. Создайте новый подпункт Главного меню, с именем Графический растровый редактор.
12. Установите для мыши шлейф.
13. Включите заставку:  
МЕТАМОРФОЗЫ, параметры - пружина, клетка, время - 1 минута.  
ОБЪЕМНЫЙ ТЕКСТ, текст «...», параметры по желанию.
16. Выберите цветовую гамму оформления ель, клен, пустыня, стандартная.
17. Просмотрите предлагаемые рисунки и узоры для Рабочего стола.

### **Работа с файлами, папками, ярлыками. Архивация.**

#### **Задания:**

1. На диске С: создайте текстовый файл СКАЗКА с текстом:

***Сжатие информации**—это процесс преобразования информации, хранящейся в файле, к виду, при котором уменьшается избыточность в ее представлении и соответственно требуется меньший объем памяти для хранения.*

2. В своей папке создайте папку LEGENDA.

3. В этой папке создайте текстовый файл БЫЛИНА с текстом:

***Архивный файл**—это специальным образом организованный файл, содержащий в себе один или несколько файлов в сжатом или несжатом виде и служебную информацию об именах файлов, дате и времени их создания или модификации, размерах и т.д.*

4. В папке LEGENDA создайте папку с именем МИФ.

5. В созданной папке создайте текстовый документ БАСНЯ с текстом:

***Архивация (упаковка)**—помещение исходных файлов в архивный файл в сжатом или несжатом виде.*

6. Для папки МИФ создайте ярлык на Рабочем столе. Посмотрите размер этой папки.

7. Смените значок ярлыка папки МИФ.

8. Откройте документ БЫЛИНА и допишите текст:

***Разархивация (распаковка)**—процесс восстановления файлов их архива точно в таком виде, какой они имели до загрузки в архив. Сохраните документ, как БЫЛИНА 1 в папке МИФ.*

9. Скопируйте документ СКАЗКА в папку МИФ. Переименуйте папку МИФ, дав новое имя, РЕКЛАМА.

10.С помощью программы–архиватора WinZip создайте архив всех файлов, находящихся в папке РЕКЛАМА.

11 .Создайте копию архивного файла под другим именем на диске А. 12.Переместите папку LEGENDA на диск А.

13.Распакуйте, архив на диске А. Переместите файлы в свою папку с номером группы. группы.

14.С помощью программы - архиватора WinRar создайте архив файла СКАЗКА, и поместите в папку LEGENDA.

15.Отчитайтесь по проделанной работе преподавателю.

## **MS – DOS**

### **Лабораторная работа №1.**

#### **Работа с файлами.**

1. Создайте текстовый файл file.txt, содержащий следующий текст:

Файл-это: одна программа; одна картинка – короче говоря, единица народонаселения компьютерного государства.

2. Создайте текстовый файл sostav.txt, содержащий следующий текст:

Основные составные части MS DOS:

1. BIOS;
2. Ядро DOS;
3. Дисковые файлы
4. Командный процессор DOS
5. Драйверы устройств

3.Переименовать файл file.txt в файл prog.txt

4.Переименовать файлы sostav.txt и prog.txt таким образом, чтобы расширение изменилось на .doc

5.Скопировать файл sostav.txt в файл prog.txt

6.Просмотреть содержимое этих файлов.

7.Скопировать файл prog.txt в файл part.txt

8.Сравнить содержимое трех имеющихся файлов.

9.Удалить файлы sostav.txt, part.txt, prog.txt.

#### **Работа с каталогами**

1. Просмотрите список файлов и подкаталогов каталога С, используя команду DIR

2. Просмотрите зарегистрированное в системе время, дату.

3. Создайте на диске С подкаталог KOT, проверьте его наличие на диске (используйте команду DIR)
4. В каталоге KOT создайте подкаталог MOUSE, проверьте его наличие
5. В каталоге MOUSE создайте файл MAK.txt со словами:

Жили у бабуси два веселых гуся,  
Один серый, другой белый – два веселых гуся.
6. В каталоге MOUSE создайте файл MUSA.txt со словами:

Мороз и солнце  
День чудесный.
7. Просмотрите содержимое файлов MAK.txt, MUSA.txt
8. Создайте копию файлов MAK.txt и MUSA.txt в файле с именем TANK.txt в каталоге KOT, проверьте его записи в нужный каталог, просмотрите содержимое файла.
9. Переименуйте файл MAK.txt в файл DOG.txt
10. Войдите в каталог MOUSE, переименуйте файл MUSA.txt в файл LOR.txt, проверьте правильность выполнения задания.
11. Удалите все созданные вами файлы и каталоги.

### **Лабораторная работа №2**

#### **Работа с каталогами и дисками**

1. В корневом каталоге диска С: создайте каталог **STUDENT**.
2. В каталоге **STUDENT** создать каталог **GROUP**.
3. В каталоге **STUDENT** создать два файла. Файл a1.txt, содержащий текст: *ИНФОРМАТИКА - это наука*. Файл a2.txt, содержащий следующий текст: *Я умею работать в среде WINDOWS*.
4. Просмотреть каталог **STUDENT**.
5. В каталоге **GROUP** создать каталог **DISK**.
6. В каталоге **DISK** создайте 2 файла. Файл **каталог.txt**, содержащий следующий текст: *Текущим называется каталог, с которым в данный момент работает пользователь*. Файл **путь.txt**, содержащий следующий текст: *Путь - это последовательность имен каталогов, разделенных символом « \ ». Этот путь задает маршрут к тому каталогу, где находится нужный файл..*
7. Просмотреть содержимое файлов **каталог.txt** и **путь.txt**.
8. Создайте точную копию файлов **каталог.txt** и **путь.txt** в файле с именем **итог.txt** в каталоге **STUDENT**.
9. Просмотреть содержимое файла **итог.txt**.
10. Скопировать файл **a1.txt** в каталог **GROUP**.

11. Просмотреть каталог **GROUP**.
12. Перейти на диск **A:**
13. На диске **A:** создать каталог **INFO**.
14. В каталоге **INFO** создать файл **disk.txt**, содержащий следующий текст: *Дискета 3.5 дюймов вмещает около 1.44 Мб информации.*
15. Скопировать файл **a2.txt** в каталог **INFO** с именем **win.txt**.
16. В каталоге **INFO** создать каталог **CLASS**.
17. Просмотреть каталог **INFO**.
18. Скопировать файл **итог.txt** в каталог **CLASS**.
19. Скопировать содержимое каталога **INFO** в каталог **GROUP**.
20. Переименовать файл **a1.txt** в каталоге **GROUP** в файл **disk1.txt**
21. Просмотреть каталог **GROUP**.
22. Просмотреть оглавление диска **A:**
23. Показать выполненную работу преподавателю.
24. Удалите все созданные вами файлы и каталоги.

### **Total Commander** **Лабораторная работа №1.**

ТЕМА: Работа с файлами и каталогами в среде TOTAL COMMANDER.

ЦЕЛЬ: Изучить режимы работы TOTAL COMMANDER: редактирование, копирование, перенос, переименование файла; создание, удаление файла и каталога.

#### **Задание:**

1. Найдите на экране: строку состояния, командную строку, строку-подсказку, объясните их назначение.
2. В корневом каталоге диска C: создайте два каталога DOG и GRAND.
3. В каталоге DOG создайте файл flora.txt, содержащий текст:  
*Ярлык—это маленький файл указатель.*
4. Выпишите в тетрадь полную информацию о файле florfa.txt.
5. Скопировать файл flora.txt в каталог GRAND.
6. В каталоге GRAND переименовать файл flora.txt на fauna.txt.
7. В файле fauna.txt допишите:  
*Файл—это поименованная область на диске.*
8. Перенести файл fauna.txt в каталог DOG с именем itog.txt.
9. Просмотреть содержимое файла itog.txt.
10. Переименовать каталог DOG в DATA.

11. Покажите преподавателю свою работу.

12. Удалить созданные каталоги.

### **Лабораторная работа № 2.**

ТЕМА: Основные комбинации клавиш.

ЦЕЛЬ: Изучение комбинаций клавиш Ctrl, Alt, Shift с клавишами {F1}-{F10}.

#### **Задания:**

1. На правой панели просмотрите информацию о текущем диске.

2. В корневом каталоге диска C: создайте каталог с именем BLANK.

3. В каталоге BLANK создайте файл tab.txt, содержащий текст:

***Выделение**—это способ указания обрабатываемой программе, какие файлы или каталоги подлежат последующей обработке.*

4. Скопируйте файл tab.txt в файл tab2.txt каталога BLANK.

5. Проверьте, есть ли в файле tab2.txt текст (если нет, повторить 4-й пункт).

6. Переименуйте файл tab2.txt в otchet.txt.

7. Приведите файл otchet.txt к следующему виду:

***Выделение**—это способ указания обрабатываемой программе, какие файлы или каталоги подлежат последующей обработке. Выделение приводит к изменению цвета соответствующей строки панели и отличает ее от других невыделенных строк.*

8. Создайте каталог KONTORA в каталоге DAR.

9. В каталоге KONTORA создайте 2 файла:

kin.bak с текстом:

***Шрифт** - это набор буквы цифр, специальных символов и знаков препинания, отображаемых определенным образом;*

don.txt с текстом:

***Колонтитулы** - элемент оформления документа, расположенный в верхнем и/или нижнем полях страницы.*

10. В каталоге DAR создайте файл kad.doc с текстом:

*Наилучшим способом хранения упорядоченной информации является таблица. В таблице могут храниться текстовые данные, но и графики, диаграммы и рисунки*

11. Перенесите файл kin.bak в каталог DAR.

12. Перенесите каталог BLANK в каталог KONTORA.

13. Осуществите быстрый поиск файлов:

- otchet.txt
- файлы, начинающиеся с символа f
- все файлы, имя которых не больше 4-х символов, расширение .exe.

- все файлы, имя которых не больше 6-и символов.
- Все файлы начинающиеся с буквы p и расширением .txt

13. Скопируйте в каталог BLANK файлы, указанные преподавателем.
14. В каталоге BLANK выделите все файлы при помощи клавиши INS.
15. Посмотрите, сколько места на диске занимает суммарная длина файлов. ! 6. Удалите каталоги KONTORA, BLANK, DAR,
17. На левой панели вызовите диск C:.
18. Для диска C: измените порядок вывода информации о файлах (сортировать): по имени; по расширению; по времени; по размеру; не сортировать оглавление каталога.
19. Отчитайтесь о проделанной работе преподавателю.

### **Текстовый редактор Word** **Лабораторная работа №1.**

Задания:

1. Запустите текстовый редактор Word.
2. Установите следующие параметры страницы: Верхнее - 2,0 см; Нижнее - 2,5; Левое-3,0см; Правое- 1,0см. Задать ориентацию (произвольно).
3. Набрать в нем текст, используя для вставки даты и времени пункт меню Вставка →Дата и Время.

#### ПРИГЛАШЕНИЕ

Уважаемый господин,

Соколов Геннадий Петрович!

Приглашаем Вас на общее собрание общества «Союз студентов России». Будем рады видеть Вас и Ваших друзей, которые хотели бы познакомиться с членами нашего общества и принять участие в обсуждении вопросов. На вечере Вы сможете посмотреть студенческое шоу и посетить студенческий бар.

Президент общества «Союз студентов России»

Молодцов В.К.

25 октября 2003г.

4. Сохранить текст в файле с именем text1.doc
5. Закрыть окно, в котором содержится этот текст.
6. Открыть новое окно и набрать в нем следующий текст, установив полуторный интервал между строками (Формат→Абзац):

Я хочу научиться работать на персональном компьютере. Для этого в частности, надо познакомиться с одним из текстовых редакторов. Я выбираю Word. Для получения навыков работы с Word я, пользуясь клавиатурой, набираю текст и работаю с ним.

7. Убрать в четвертом предложении фразу «пользуясь клавиатурой».
8. В первом предложении выделить красную строку.
9. Вставить заголовок «ТЕКСТ №1».
10. Скопировать первое предложение в конец текста.
11. Открыть документ text1.doc
12. Оформить заголовок текста ПРИГЛАШЕНИЕ полужирным шрифтом размером 14 пт.
13. Оформить фрагмент текста с ФИО полужирным курсивом размером - 12 пт.
14. Выделить разными шрифтами должность и фамилию приглашающего, дату отправления
15. Скопировать «ТЕКСТ №1» в конец документа text1.doc
16. Проверить грамматику.
17. Удалить текст приглашения.
18. Отменить данное действие.
19. Найти в тексте все слова, начинающиеся на букву «W».
20. Заменить слово «студентов» на «пенсионеров», используя, пункт меню Правка.
21. Скопировать текст из окна два раза (в этом же окне), озаглавив их соответственно «ТЕКСТ №2» и «ТЕКСТ №3».
22. «ТЕКСТ №1» выровнять по левому краю «ТЕКСТ №2» - по центру, «ТЕКСТ №3» - по ширине, с помощью панели форматирования.
23. Осуществить предварительный просмотр документа.
24. Сохранить все документы.

### **Лабораторная работа №2.**

Задания:

1. Загрузите текстовый редактор Word.
2. Создайте таблицу из 4 столбцов и 17 строк, заполните ее по описанию:  
Первая строка - заголовок таблицы (все столбцы объединены);  
**Прежде чем форматировать таблицу или группу ячеек (обрамление, заливка, объединение ячеек и т.д.) необходимо применить выделение.**  
Вторая строка - невидимая;  
С третьей по шестнадцатую строку заполнить по образцу.

Отчет о доходах и расходах предприятия «Восход»

Показатель	месяц		
	Июнь	Июль	Август
Валовая выручка	150,16	169,95	198,82
Налог на добавленную стоимость			
Выручка от реализации			
Затраты на производство продукции	90,86	120,99	157,16
В том числе: заработная плата	600	600	600
Арендная плата	2,5	2,5	2,5
Аренда оборудования	1,5	1,5	1,5
Валовая прибыль			
Облагаемый доход			
Налог на доход предприятия			
Чистый доход предприятия			

3. Ввести формулу для расчета:

- Налога на добавленную стоимость (НДС): Валовая выручка \* коэффициент НДС (20%);
- Выручка от реализации: Валовая выручка - НДС;
- Валовая прибыль: Выручка от реализации - Затраты на производство продукции;
- Облагаемый доход: сумма валовой прибыли и заработной платы;
- Налог на доход предприятия: 22% от облагаемого дохода;
- Чистый доход предприятия: валовая прибыль - налог на доход предприятия.

4. Вставить после таблицы формулы:

$$\text{a) } F(x) = \frac{1}{\delta \cdot \sqrt{2\pi}} \int_{-\infty}^x \varepsilon^{\frac{1}{2} \left( \frac{x-a}{\delta} \right)^2} dx ; \quad \text{б) } \square = \iint \frac{\frac{M}{\Omega^2} d\Omega}{\sqrt{2\eta[E - K(\Omega)] - \frac{M^2}{\Omega^2}}} + const .$$

5. Преобразовать формулу  $F(x)$  к виду:

$$F(x) = \frac{1}{\delta \cdot \sqrt{2\pi}} \int_{-\infty}^x \varepsilon^{\frac{1}{2} \cdot \frac{1}{\delta^2} \left( \frac{x-a}{\delta} \right)^2} dx$$

6. Перенесите формулы на вторую страницу.

7. Скопировать таблицу на третью страницу.

8. Расставить номера страниц и задать колонтитул:

- Для первой страницы «Данный документ подготовлен в редакторе Word»;
- Для четной страницы «Лабораторная работа №2»;
- Для нечетной страницы «Документ создан ... (вставить текущую дату)».

9. Сохранить текст в файле tab.doc

**Лабораторная работа №3.**

Задание 1:

Наберите текст и расположите его в трех колонках, как показано на образце. Воспользуйтесь командой **Формат→Колонки**.

**Не бойся попробовать**

Лучший способ научиться проектировать документы – больше практиковаться в этом. Это упражнение дает вам возможность, используя современный, хорошо развитый текстовый процессор, научиться использовать основные приемы и процедуры, принятые в издательских системах.

**Попробуй еще раз**

Здесь имеется большое количество возможностей для экспериментирования. Начнем с выбора шрифтов, их типа, размера и стиля. Затем научимся располагать текст документа в колонках, используя колонки разной ширины и газетные колонки.

**И еще раз**

Не следует забывать и об использовании различных способов выделения, таких как жирный и курсив, подчеркивание, выделение части текста в рамку и т.д.

## Лабораторная работа №4.

Задание 1:

1.Создайте следующий список (маркированный, нумерованный, многоуровневый), соблюдая все элементы форматирования:

- Заголовки выровнены по центру, размер шрифта (18 пт), полужирный, курсив.
- Маркированный список - размер шрифта(14 пт), курсив, цвет шрифта коричневый, выровнен по левому краю.
- Нумерованный список - размер шрифта(12 пт), курсив, цвет шрифта сиреневый, подчеркнутый, выровнен по центру.
- Нумерованный список - размер шрифта(16 пт)/ курсив, цвет шрифта синий, контур выровнен по правому краю.
- Маркированный список – размер шрифта(14 пт), курсив, цвет шрифта фиолетовый, выровнен по левому краю, двойное зачеркивание.
- Многоуровневый список – размер шрифта(18 пт), курсив, цвет шрифта зеленый, выровнен по левому краю.

### ***Системы счисления:***

- *Понятие системы счисления*
- *Непозиционные системы счисления*
- *Позиционные системы счисления'*

### ***Базовая структура персонального компьютера:***

*I. Элементы системного блока*

*II. Микропроцессор*

*III. Блок питания*

*IV. Контроллеры*

### ***Компьютерные сети:***

*a) Локальные компьютерные сети*

*b) Глобальные компьютерные сети*

### ***Работа в сети:***

~~● *Электронная почта*~~

~~● *Почтовые серверы – файлы почтой*~~

- ~~Word Wide Web — паутина, которая соединяет~~
- ~~Работа с FTP~~

## **Массивы и пользовательские типы данных**

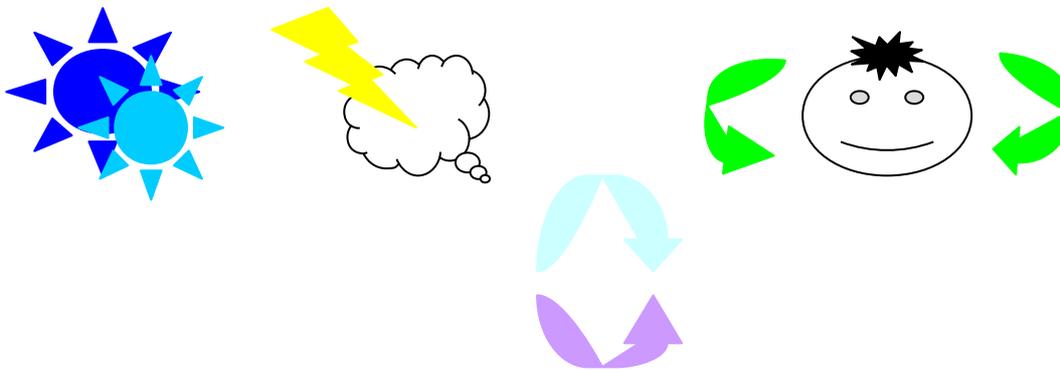
*a.1.1.1.1. Одномерные массивы*

*a.1.1.1.2. Двумерные массивы*

*a.1.1.1.3. Символьные массивы*

Задание 2:

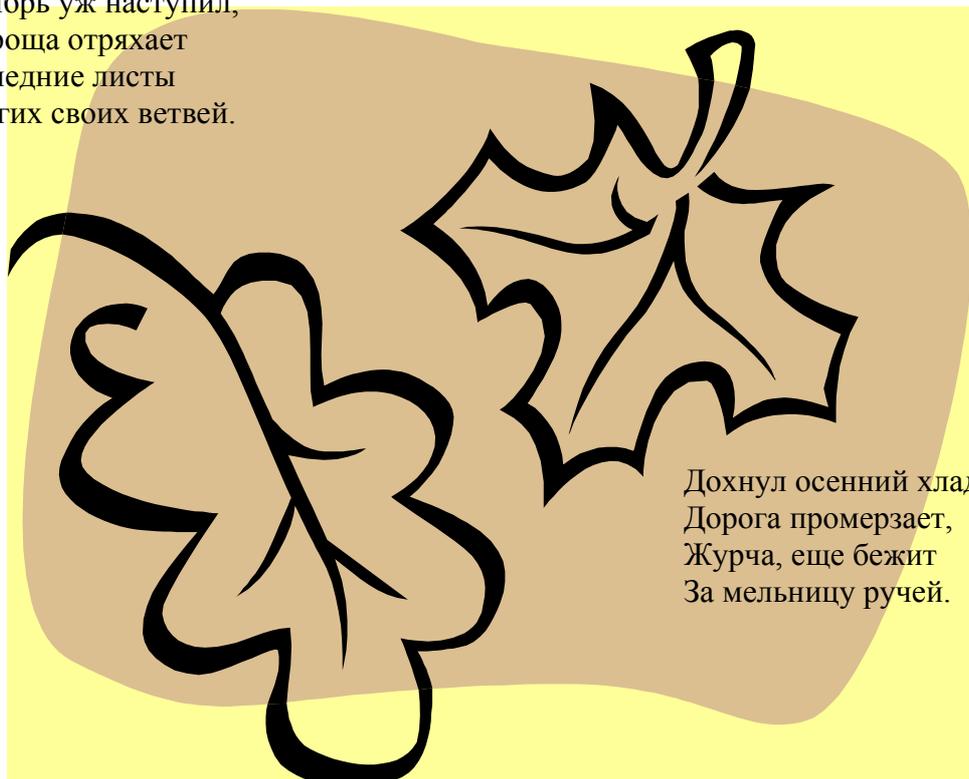
Используя автофигуры, создайте следующие объекты.



Задание 3:

Оформите фрагмент стихотворения по образцу.

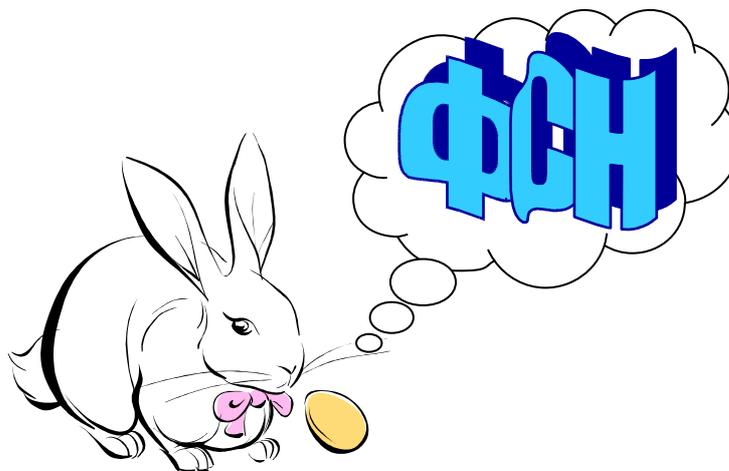
Октябрь уж наступил,  
Уж роща отряхает  
Последние листья  
С нагих своих ветвей.



Дохнул осенний хлад,  
Дорога промерзает,  
Журча, еще бежит  
За мельницу ручей.

Задание 4.

Оформите текст по образцу. Используйте объект **WordArt**.



### Excel

#### Задание 1:

Создайте прейскурант для пересчета цен товаров в рублях на доллары в соответствии с текущим курсом доллара.

Прейскурант		
Курс пересчета	1 у. е.	
Наименование товара (10)	Цена руб.	Цена (у. е.)

Лист1 переименуйте в Задание1.

Заполните таблицу данными: Наименование товара, цена, курс доллара. Рассчитайте «Цена(у.е)», используя в формуле абсолютную ссылку.

Задайте форматирование таблицы.

Добавьте одну строку для заголовка. Придумайте заголовок для таблицы.

#### Задание2:

Рабочий лист должен содержать сводную ведомость студенческих оценок по итогам сессии. Если экзамены сданы без троек, то соответствующая ячейка таблицы должна подсвечиваться зеленым цветом, если у студента остались задолженности – красным.

1. Выберите новый рабочий лист и переименуйте его как «Ведомость».

2. В первую

строку введите:

Ф.И.О.	Математика	Информатика	История	Средний балл	Стипендия	Доплата
			109			
1-						
10						

3. Заполните таблицу. Экзаменационные оценки должны изменяться от 2 до 5 баллов
4. В ячейку G2 введите значение фиксированной доплаты- 100р.
5. Посчитайте средний балл каждого студента.
6. Посчитайте стипендию: если студент имеет средний балл больше или равный 4, то стипендия равна 400 руб. + фиксированная доплата, иначе 200 руб. + фиксированная доплата.
7. В конце ведомости посчитайте количество 5,4,3,2 по каждому предмету.
8. Вычислите средний балл группы.
9. Задайте условное форматирование для студентов: (Фор мат→Условное форматирование).  
Если оценка больше 3 то фон ячейки – светло-зеленый; а если оценка меньше или равна 3, то фон ячейки - красный.
10. По данным ведомости построить гистограмму.
11. Для предмета «Информатика» постройте круговую диаграмму.

### **Задание 3:**

Перейдите на Лист 3. Переименуйте его в Задание 3. В ячейках A1,B1,и C1 поместите 3 произвольных числа.

- Сумма этих чисел превышает их произведение, то в ячейку D1 выведите абсолютное значение разности содержимого этих ячеек, в противном случае- корень из их произведения.
- Модуль разности этих чисел превышает 20, то в ячейку D1 выведите их среднее арифметическое, в противном случае- их произведения.
- Если модуль их суммы меньше либо равен их среднему арифметическому, то в ячейку D1 выведите их среднее геометрическое, в противном случае – сумму квадратов этих чисел.
- Если среднее арифметическое первых двух чисел больше третьего, то в ячейку D1 выведите слово «Больше», иначе «Меньше».
- Если их среднее арифметическое превышает их разность, то в ячейку D1 выведите их среднее геометрическое, в противном случае- абсолютное значение их суммы.

### **Задание 4:**

Перейдите на Лист 4.

Переименуйте его в Задание 4.

Постройте точечную диаграмму функций:

$y=x^2$  и  $y=x^3$  в интервале от -4 до 4 с шагом 0,1 ;

$y=2x^2-2$  в интервале от -3 до 3 с шагом 0,5;

$y=\cos x$  в интервале от 0 до 4 с шагом 0,2;

### Задание 5:

#### Создание диаграмм

1. Перейдите на новый лист на Лист 5 или добавьте новый лист
2. Назовите лист – Продажи за 2000 г
3. Введите таблицу, рассчитайте значения Итого.

Продажи за 2000				
	1 квартал	2 квартал	3 квартал	4 квартал
Магазин 1	117,5	101,8	125,3	140,5
Магазин 2	73,5	81,3	90,7	101,1
Магазин 3	84,3	96,1	89,1	98,3
Итого				

4. Для «Магазина 1» постройте круговую диаграмму на отдельном листе.
5. Постройте гистограмму по продажам все магазинов.
6. Измените название гистограммы на «Сводка продаж фирмы «Альфа» за год».
7. Измените тип диаграммы на «Трубчатая».

### Задание 6:

#### Обработка информации в списках.

1. Перейдите на новый лист
2. Создайте таблицу-список по образцу:

Представитель	Регион	Месяц	Объем	Описание
Иванов	Запад	Январь	\$500	Звуковые карты
Петров	Север	Январь	\$100	Звуковые карты
Сидоров	Юг	Февраль	\$600	Звуковые карты
Андреев	Восток	Февраль	\$200	Звуковые карты
Петров	Север	Январь	\$1500	CD-ROM
Сидоров	Юг	Февраль	\$1100	CD-ROM
Петров	Север	Февраль	\$800	CD-ROM
Иванов	Запад	Февраль	\$1200	Принтеры
Петров	Север	Январь	\$2500	Принтеры

Выделите заголовки столбцов жирным шрифтом, числовые данные в столбце Объем должны иметь Денежный формат, в долларах США. Примените ко всей таблице автоформат Цветной2.

3. Форма для ввода данных.

Для ввода данных с помощью формы выделите любую ячейку внутри таблицы и выполните команду меню Данные→Форма. В появившейся форме нажмите, кнопку Добавить.

Занесите еще две записи в список:

Сидоров	Юг	Январь	\$120	Звуковые карты
Петров	Север	Февраль	\$320	CD-ROM

#### 4. Проверка данных при вводе.

- Выделите ячейки столбца «Месяц», для которого устанавливается проверка ввода (в диапазон должны войти только названия месяцев, а также несколько пустых ячеек, расположенных ниже).
- Выполните команду меню Данные→Проверка. Перейдите на вкладку «Параметры».
- Установите условие проверки для типа данных «Дата», значения между 01.01.2000 и 01.02.2000.
- Перейдите на вкладку «Сообщение об ошибке», в поле «Сообщение», введите текст "Неправильная дата".
- Попробуйте ввести в столбец Месяц неправильную дату.

#### 5. Сортировка строк и столбцов.

- Выделите любую ячейку в столбце Объем.
- Выполните команду Данные→Сортировка.
- В списке «Сортировать по...» выберите заголовок первого столбца и тип сортировки «По убыванию».

#### 5. Автофильтр.

–Выберите любую ячейку в списке. Выполните команду Данные→Фильтр→Автофильтр. (В верхней ячейке каждого столбца появляется раскрывающийся список.) Из раскрывающегося списка в столбце Представитель выберите фамилию Петров. В таблице отобразятся только сведения о его продажах. Чтобы вернуться к полному списку, из раскрывающегося списка выберите (Все).

–Для того чтобы отфильтровать список по нескольким представителям, выберите из списка (Условие)., В появившемся окне в выберите равно Петров, поставьте переключатель ИЛИ, ниже выберите равно Сидоров. Отсортируйте полученный список по столбцу Представитель по возрастанию.

–В столбце Объем задайте условие, при котором бы отображались объемы продаж на сумму более 120\$. Добавьте к условиям сортировки (Данные→Сортировка) по Объему по возрастанию.

–Установите условие выбора для отображения сведений о продажах представителя Петров за январь.

–Отмените все условия сортировки, выбрав из списка в столбце Представитель – (Все) и из списка в столбце Объем–(Все).

#### 7. Вывод промежуточных итогов.

Отсортируйте список по столбцу «Представитель»

Выполните команду Данные→Итоги. Для вывода данных об общей сумме продаж для каждого представителя в появившемся диалоговом окне установите следующие параметры):

«При каждом изменении в: Представитель»

«Операция: Сумма»

«Добавить итоги по: Объем»

«Заменить конечные итоги»

«Итоги под данными».

Выделите цветной заливкой строки Всего по каждому представителю и Общий итог.

## **СУБД ACCESS**

### **Лабораторная работа №1**

#### **Создание собственной базы данных**

##### **1. Мастер таблиц.**

Рассмотрим задачу, связанную с назначением на стипендию по результатам экзаменационной сессии. Стипендия назначается студентам, получившим на экзаменах хорошие и отличные оценки. Если хороших оценок получено более одной, назначается стипендия в размере 100%. За одну хорошую и все остальные отличные оценки устанавливается стипендия в размере 150%, за все отличные 200%.

Данные организованы в таблицы: СТУДЕНТ, СЕССИЯ, СТИПЕНДИЯ.

Таблица СТУДЕНТ содержит необходимые сведения о каждом студенте, обучающемся в вузе. Таблица СЕССИЯ содержит сведения о результатах сдачи студентами четырех экзаменов. Таблица СТИПЕНДИЯ содержит информацию об условиях назначения студентов на стипендию.

1. Создайте БД и сохраните ее с именем SESS.mdb.
2. В окне БД выберите страницу ТАБЛИЦЫ. Щелкните на кнопке СОЗДАТЬ, Выберите в списке Мастер таблиц и щелкните на кнопке ОК. Вы увидите первое окно Мастера таблиц (рис. 1). В списке слева перечислены примеры таблиц. Под списком расположены кнопки Деловое применение и Личное применение. Содержание списка зависит от вашего выбора.

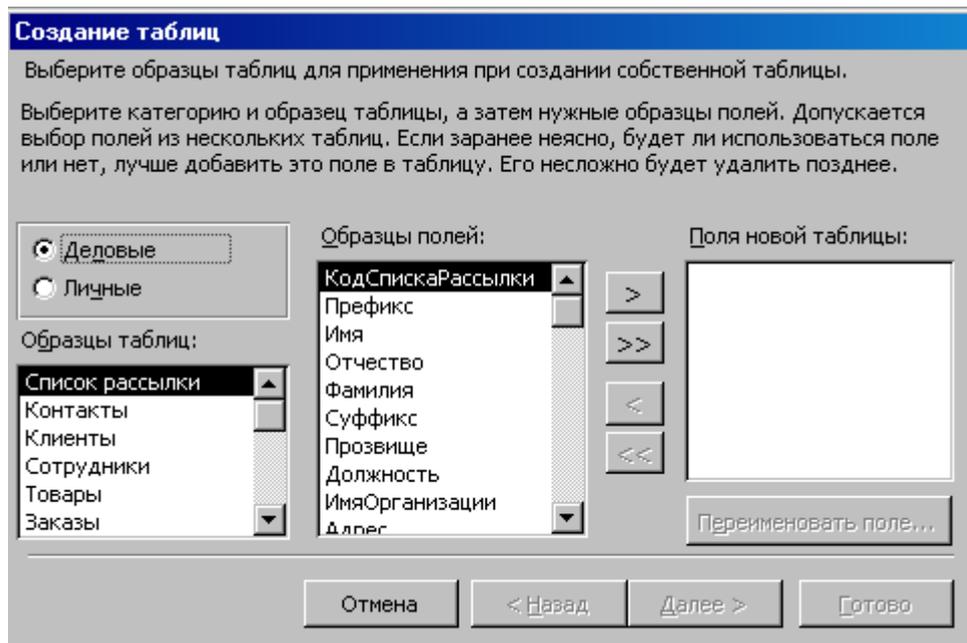


Рис.1

Выберите Личное применение, чтобы просмотреть таблицы, предназначенные для личного пользования. Для нашей задачи выберите Деловое применение и в списке образцы таблиц – таблицу «Студенты». При выборе таблицы в списке в середине появится перечень предлагаемых полей. Для перемещения нужного поля из среднего списка в список слева, щелкните на имени поле и затем на кнопке >. Если вам не нравится какое-либо название, его можно изменить: выделите поле и щелкните на кнопке переименовать.

3. Поместите в список слева поля: фамилия, имя, отчество, город, группа (выберите поле номер студента и переименуйте). Нажмите кнопку Далее. Откроется второе окно Мастера таблиц, в котором задается имя таблицы и определяется ключевое поле в новой таблице. Задание ключа может идти несколькими путями. Если выбран режим автоматического определения ключа и в новую таблицу включено поле счетчик, то последнее выбирается в качестве ключевого. Второй путь – задание ключевого поля самостоятельно.
4. Задайте имя таблицы «Студенты» и установите самостоятельное определение ключа. Определите ключевое поле НОМЕР.

На третьем шаге Мастер таблиц предоставляет возможность связать новую таблицу с другими таблицами БД. Список существующих таблиц БД выводится в окне. Если в какой-нибудь из таблиц есть поле, совпадающее с ключом создаваемой таблицы, Access предложит наличие связи. С выбором Access можно согласиться, отказаться или создать связь вручную. Если создаваемая таблица первая в БД, то Мастер таблиц пропускает третий шаг.

На последнем шаге пользователю остается определить режим, который будет активизирован после завершения работы Мастера таблиц. Переключатель Изменение структуры

таблицы означает переход в режим Конструктора для новой таблицы. Выберите его, если необходимо доделать то, что не смог выполнить Мастер таблиц: ввести новые поля, придать им необходимые свойства, переопределить ключ и т. д. Переключатель непосредственный ввод данных в таблицу приведет к тому, что таблица будет открыта для просмотра в табличной форме. Переключатель ввод данных в таблицу с помощью формы заставит Мастер таблиц создать форму для новой таблицы.

5. Завершите работу Мастера таблиц с последующим переключением в режим Конструктора для добавления новых полей.

### Режим Конструктора.

В режиме конструктора можно не только вводить имена полей, но также выбирать их тип и задавать их свойства. Окно конструктора представлено на рис. 2.

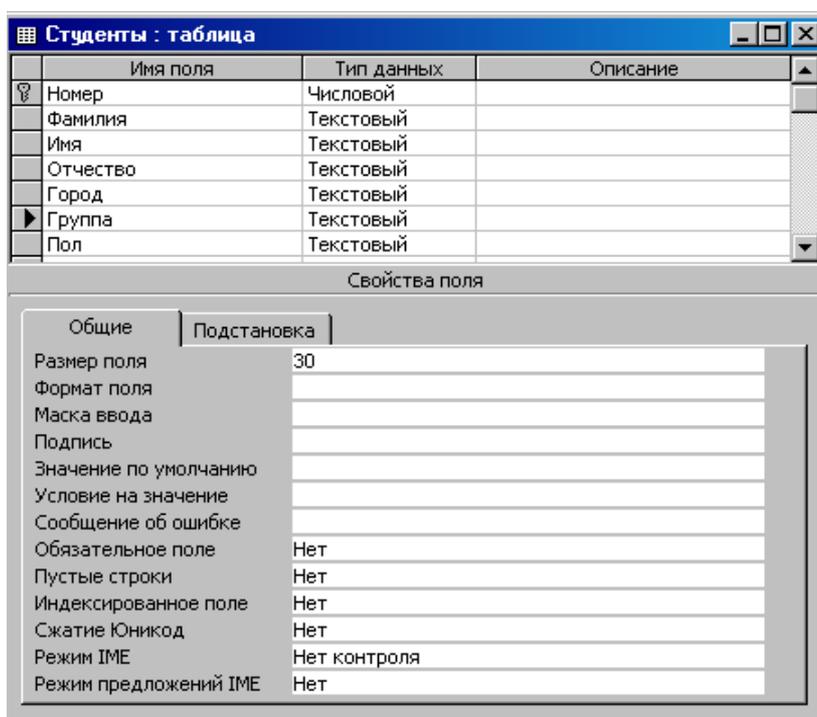


Рис. 2

### Типы полей доступные в Конструкторе:

**Текстовый:** используется для полей, не участвующих в вычислениях, размер не превышает 255 символов.

**Поле МЕМО:** используется, когда необходимо ввести более 255 символов.

**Числовой:** используется в случаях, когда необходимо производить вычисления.

**Дата/время:** с его помощью можно выполнять действия над датами.

**Денежный:** числовой тип с фиксированным числом разрядов после запятой и указанием денежной единицы.

**Счетчик:** используется для автоматической нумерации записи.

**Логическое:** поле снабжено флажком. Установленный флажок означает – Да, сброшенный – Нет.

**Поле объекта OLE:** используется для рисунков, звуковых файлов или графиков.

**Мастер подстановок:** используется для создания поля, которое дает возможность выбирать значения из полей в другой таблице.

### **Свойства полей.**

**Формат поля:** доступен для\_числовых, денежных полей, поля дата/время для определения формата вывода на экран.

**Маска ввода:** задает маску ввода, облегчающую ввод данных в элемент управления – поле. Например, удобно создать следующую маску ввода для поля «Телефон», позволяющую вводить только цифры и автоматически добавляющую промежуточные символы: (\_\_\_\_)\_\_\_\_ - \_\_\_\_\_. Значение данного свойства определяется автоматически при использовании мастера по созданию масок ввода. Символы, которые следует вводить в маску ввода, определяются следующими специальными символами:

**0** Цифра (обязательный символ; знаки (+) и (-) не разрешены).

**9** Цифра или пробел ( не обязательный символ; знаки (+) и (-) не разрешены)

**#** Цифра или пробел ( не обязательный символ; незаполненные позиции выводятся как пробелы в режиме редактирования, но удаляются при сохранении данных; знаки (+) и (-) не разрешены)

**L** Буква (обязательный символ)

**?** Буква (необязательный символ)

**A** Буква или цифра (обязательный символ)

**a** Буква или цифра (необязательный символ)

**&** Любой символ или пробел (обязательный символ)

**C** Любой символ или пробел (обязательный символ)

**.,:;-/** десятичный разделитель, разделитель групп разрядов, времени или даты. (Используемые символы разделителей определяются настройками , выбранными в окне в окне Языки стандарты панели управления Windows)

**<** Преобразует все символы к нижнему регистру.

**>** Преобразует все символы к верхнему регистру.

**!** Указывает, что маска ввода заполняется справа налево: этот символ следует использовать, если в левой части маски находятся позиции, которые не обязательно заполнять. Маски ввода обычно заполняются слева направо. Символ восклицательного знака можно помещать в произвольную позицию в маске ввода.

**\** Указывает, что следующий символ следует воспринимать как постоянный (а не специальный) символ (например, \A представляет символ «А»)

Подпись задает заголовок столбца при выводе таблицы.

**Значение по умолчанию** позволяет указать значение, автоматически вводящееся в поле при создании новой записи, но доступное изменению. Например, в таблице «Адреса» может оказаться

удобным указать автоматический ввод значения «Москва» в поле «Город». **Запись =DATE** – специальная функция, вставляющая в поле текущую дату.

**Условие на значение** определяет требование к данным, вводимым в запись, в поле или в элемент управления. Для этого используются операторы:

<>, <=>, =, <> - для проверки численного значения;

**in** (знач1, знач2,...) – для проверки включения значения в список;

**between...and** – для проверки вхождения в интервал значений (например, between 5 and 8.2)

**and, or** для связки условий.

Для сравнения дат используется символ # , например, >#0.22.99 (дат ранее 22 октября 1999 в вашей таблице не будет).

Для текстовых полей можно проверять позиции конкретных символов или ограничить их набор.

Для этого служат операторы:

? – любой символ, но только один;

# - цифра;

\* - произвольное количество любых символов;

**LIKE** – вводимое значение сравнивается с образцом.

**LIKE “# # # #” OR LIKE “ # # # # - # # # #”** означает, что введенное значение должно состоять либо из пяти цифр, либо из пяти цифр, дефиса и еще четырех цифр.

Например, условие – “LIKE [АБВГ][1234] - # # # #”.означает, что введенное значение должно на первой позиции содержать букву А, Б, В или Г, Во второй цифру 1, 2, 3 или 4, далее дефис и четырехзначное число. В результате получится значение вроде А1 – 7654. Чтобы указать значения вводить, которые категорически запрещается, нужно поставить перед ними восклицательный знак. Условие - LIKE [!ЭЮЯ] позволяет ввести любой символ кроме Э, Ю, Я.

**Сообщение об ошибке** позволяет указать текст сообщения, выводящегося на экран, если введенные данные нарушают условие, определенное в свойстве Условие на значение.

**Обязательное поле:** определяет необходимо ли вводить в поле значение.

6. В режиме Конструктора в таблицу «Студенты» добавьте следующие поля: пол (текстового типа), с указанием условия на значения «м» или «ж» и сообщением об ошибке; дата рождения (типа дата/время) с заданием маски ввода.

7. Откройте таблицу «Студенты» и заполните ее.

### **Режим таблицы.**

Режим таблицы – это превосходный способ создания простых таблиц, подходящих для ситуаций, когда вам требуется немедленно заполнить их. Создание таблицы заключается в задании полям имен и вводе данных. Окно пустой таблицы, созданной в режиме таблицы, показано на рис. 3. Для определения имени поля нужно дважды щелкнуть на Поле 1 или др. (или щелкнуть правой кнопкой мыши) и выбрать команду **Переименовать столбец**. После этого

можно вводить данные в таблицу. Столбцы таблицы можно изменять в размерах, удалять, скрывать.



Рис. 3

8. С помощью режима таблицы создайте таблицу «Стипендия», в которой содержится информация об условиях назначения студентов на стипендию.

9. Результат «отл» (стипендия 200%) соответствует всем отличным оценкам в сессии, результат «хор!» (стипендия 150%) соответствует одной хорошей оценке и остальным отличным оценкам. Результат «хор» (стипендия 100%) ставится за две четверки и более, без наличия удовлетворительных оценок. Стипендия в размере 0% устанавливается за результат «удовл» и «неудовл». (Ключевое поле Результат).

Результат	Процент
отл	200
хор!	150
хор	100
удовл	0

Рис. 4

Создайте таблицу «Сессия» с использованием режима Конструктор. Создайте поля: Номер(ключевое поле мастера подстановок полей НОМЕР и ФИО таблицы СТУДЕНТЫ), Оценка 1 (числовой), Оценка 2, Оценка 3, Результат (мастер подстановок столбца Результаты), Стипендия (мастер подстановок столбцов «Процент» и «Результат»).

Не заполняйте таблицу «Сессия» без установления схемы данных.

#### 4. Схема данных.

После создания различных таблиц, содержащих данные, относящиеся к различным аспектам базы данных, разработчик должен продумать, каким образом Access будет объединять эти данные при их извлечении из базы данных. Первым шагом при этом является определение связей между таблицами. После этого становится возможным создание запросов, форм, отчетов, в которых выводятся данные из нескольких таблиц сразу.

### **Отношение «один-к-одному»**

При отношении «один-к-одному» запись в таблице **A** может иметь не более одной связанной записи в таблице **B** и наоборот.

Отношение «один-к-одному» создается в том случае, когда оба связанных поля имеют уникальные индексы. Примером такого отношения может служить связь между объектами **СТУДЕНТ** <-----> **СЕССИЯ**. При этом устанавливаются отношения между совпадающими значениями в ключевых полях, т. е. Между полями разных таблиц имеющими одинаковые имена. С ключевым полем таблицы **СТУДЕНТ**, являющимся уникальным идентификатором каждой записи (ФИО), связывается внешний ключ таблицы **СЕССИЯ**, в которой данная запись единственна.

### **Отношение «один-ко-многим»**

Связь с отношением «один-ко-многим» является наиболее часто используемым типом связи между таблицами. В такой связи каждой записи в таблице **A** могут соответствовать несколько записей в таблице **B**, а запись в таблице **B** не может иметь более одной соответствующей ей записи в таблице **A**. Отношение «один-ко-многим» создается в том случае, когда только одно из полей является ключевым или имеет уникальный индекс. Примером такого отношения может служить связь между объектами:

**СТИПЕНДИЯ**<----->**СЕССИЯ**

При этом уникальная запись «отл» таблицы **СТИПЕНДИЯ** может использоваться один и более раз в таблице **СЕССИЯ** для разных уникальных идентификаторов записей этой таблицы (т. е. Для разных фамилий студентов).

### **Отношение «многие-ко-многим»**

При отношении «многие-ко-многим» одной записи в таблице **A** могут соответствовать несколько записей в таблице **B**, а одной записи в таблице **B** несколько записей в таблице **A**. Такая схема реализуется только с помощью третьей (связующей) таблицы, ключ которой состоит из по крайней мере двух полей, которые являются полями внешнего ключа в таблицах **A** и **B**. Связь с отношением «многие-ко-многим» фактически является двумя связями с отношением «один-ко-многим» через третью таблицу, ключ которой состоит из по крайней мере двух полей, которые являются полями внешнего ключа в двух других таблицах.

Примером такого отношения может служить связь между объектами:

**СТУДЕНТ**<<----->**ПРЕПОДАВАТЕЛЬ**

10. Просмотрите схему БД, выбрав команду **Сервис/Схема БД**. Для расположения таблицы в окне **Схема данных**, выберите команду **Связи/Добавить таблицу**.

Установите связи между таблицами **СТУДЕНТ** и **СЕССИЯ**. Для этого перетащите указатель мыши от ключевого поля таблицы **СТУДЕНТ** к ключевому полю таблицы **СЕССИЯ**. В открывшемся

окне диалога «Связи» выберите «Обеспечение целостности данных», и отношение «Один-к-одному» и нажмите кнопку **Создать**.

Выбор «Обеспечение целостности данных», не позволяет добавлять в связанную таблицу записи, для которых нет соответствующих записей в главной таблице или же изменять записи в главной таблице таким образом, что после этого в связанной таблице появятся записи, не имеющие соответствующих главных записей, а также удалять записи в главной таблице, для которых имеются подчиненные записи в связанной таблице. Это вам возможность установить еще два флажка:

«Каскадное обновление данных» (при изменении данных в одной таблице они будут изменяться в другой таблице),

«Каскадное удаление связанных записей» (при удалении данных в одной таблице они будут удаляться в другой таблице).

11. Установите связи между таблицами СТИПЕНДИЯ и СЕССИЯ. Для этого перетащите указатель мыши от ключевого поля таблицы СТИПЕНДИЯ к полю РЕЗУЛЬТАТ таблицы СЕССИЯ. В открывшемся окне диалога «Связи» выберите «Обеспечение целостности данных», и отношение «один-ко-многим» и нажмите кнопку Создать.

В результате описанных действий окно «Схемы данных» приобретет вид, показанный на рис. 5.

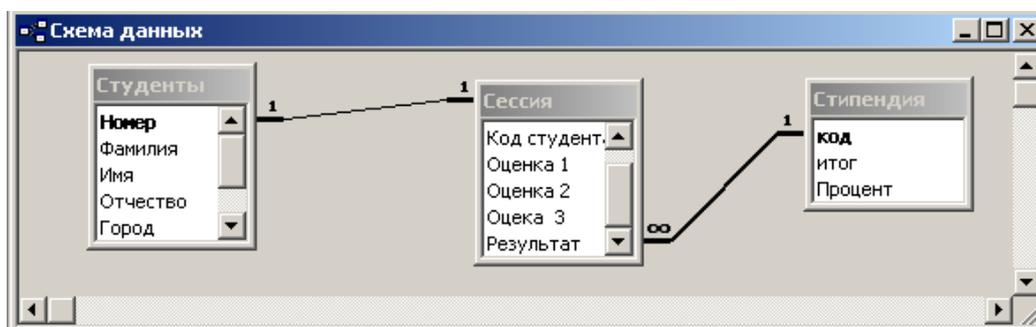


Рис. 5

11. Сохраните установленные связи между таблицами, выбрав пункт меню **Фйл/Сохранить макет**.

Заполните таблицы. Убедитесь в недопустимости повторяющихся записей ключевых полей в таблицах СТУДЕНТ и СЕССИЯ (связь «один-ко-многим»).

Измените номер зачетной книжки одного из студентов. Обратите внимание на изменение этой записи в другой таблице, при выборе каскадного обновления данных.

Удалите данные одного студента. Обратите внимание на удаление этой записи в другой таблице, при выборе каскадного удаления данных.

Проверьте наличие каскадных действий между таблицами СТИПЕНДИЯ и СЕССИЯ.

Закройте БД.

## Лабораторная работа №2

### "Формы в ACCESS" Мастер форм

В некоторых случаях удобнее вводить данные не в таблицу, а в форму, в которой каждую запись можно представить в отдельном окне.

Для исследования возможностей мастера форм используем таблицу СТУДЕНТЫ из БД sess.mdb.

1. Чтобы запустить мастер форм, перейдите на вкладку ФОРМЫ, нажмите кнопку СОЗДАТЬ, выберите МАСТЕР ФОРМ и укажите имя таблицы (СТУДЕНТЫ). Щелкните ОК. (рис. 6)

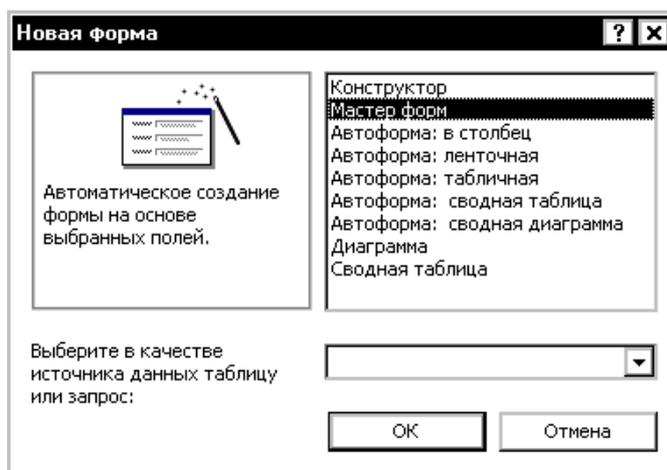


Рис. 6

Выберите все доступные поля, нажмите кнопку **Далее**. Выберите внешний вид формы **В один столбец**, нажмите кнопку **Далее**. Установите параметры оформления по своему вкусу.

В четвертом диалоговом окне задайте имя формы **ИНФОРМАЦИЯ О СТУДЕНТАХ**. Щелкните **ГОТОВО**.

Каждая запись (то, что являлось отдельной строкой таблицы) теперь представляет собой как бы отдельную карточку. Вводить данные удобнее именно в таком режиме.

В строке состояния (в нижней части окна формы) отражено общее число записей и номер записи, которую вы сейчас видите.

2. Выполните, используя форму: пролистайте все имеющиеся записи; перейдите к первой записи; перейдите к последней записи; внесите исправления в некоторые записи; добавьте две записи. Пришло время проверить, отобразились ли наши записи в исходной таблице. Для этого:

В окне **База данных** выберите вкладку **Таблицы** и откройте таблицу СТУДЕНТЫ. Если новые записи не внесены в таблицу, то выполните команду **Записи Показать все записи**. Если вы сохранили форму перед тем, как перейти к таблице, то новые записи будут внесены в таблицу автоматически.

3. Закройте таблицу и перейдите вновь к форме СТУДЕНТЫ.

## Представление формы в режиме таблицы

Форму можно представлять и в режиме таблицы командой **Вид Таблица**. В режиме таблицы также можно вносить данные, редактировать их. Можно удалять записи, если выделить строку таблицы и нажать клавишу {Delete}.

В таблице существует возможность переставлять столбцы. Сделать это можно, "схватив" столбец за заголовок мышью, перенести его на новое место

## Конструктор формы. Элементы управления.

Для создания нового поля уже в существующей форме (например, проживает в общежитии или нет) познакомимся с элементами управления.

*Элементы управления* — это графические объекты, размещенные в форме или отчете и предназначенные для изображения данных, выполнения операций или просто для красоты.

Наиболее распространенная разновидность элементов управления — текстовое поле, которое позволяет размещать текст или цифры. Его можно использовать для ввода новых данных или редактирования имеющихся. Но для выполнения поставленной задачи нас больше интересует флажок или переключатель.

Рассмотрим вид окна формы в режиме конструктора (рис. 7).

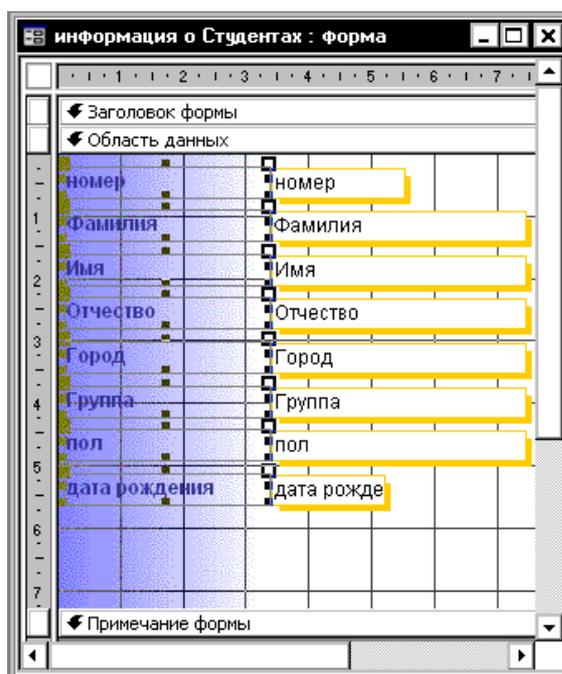


Рис. 7

Область данных	<p>Служит для размещения основных элементов управления. Может быть увеличена (или уменьшена) за счет изменения нижней границы. Именно в этой области и будет происходить дальнейшая работа.</p>
Добавление элементов управления	<p>Панель элементов позволяет добавлять элементы управления в формы или отчеты.</p> <p>Для добавления элемента управления следует выбрать соответствующий инструмент.</p> <p>При переводе в форму или отчет указатель приобретает вид крестика с расположенным рядом значком выбранного элемента управления. Требуется установить указатель в форме в позиции, в которой должен находиться верхний левый угол элемента управления, и нажать кнопку мыши. В случае отсутствия на экране панели элементов, вывести ее можно нажатием кнопки панели инструментов или вызовом команды Вид Панель элементов.</p>
Область заголовка формы	<p>Используется для вывода текста заголовка формы, инструкций по работе с формой или кнопок, открывающих связанные формы или выполняющих другие задачи. Область заголовка формы (подпись) выводится в верхней части окна формы и в начале печатной формы. Подпись не связана с полями базы данных, она является постоянной, т. е. ее содержание не меняется при перемещении по записям формы. Подпись можно изменить или удалить, выделив имеющуюся. Можно добавить новую подпись, выбрав инструмент панели элементов кнопку. Подписи можно перемещать и располагать по своему вкусу. Область заголовка формы можно расширить, если опустить нижнюю границу, "схватив" ее мышью.</p>
Область примечаний	<p>Область примечаний формы выводится в нижней части окна формы. Используется для вывода инструкций по работе с формой, кнопок или свободных элементов управления, принимающих действия пользователя. Область заголовка формы можно расширить, если опустить нижнюю границу, "схватив" ее мышью.</p>

Для вставки элемента управления предназначена панель **Элементы управление**. В случае отсутствия на экране панели элементов, вывести ее можно нажатием кнопки панели инструментов или вызовом команды **Вид Панель элементов**.



Рис. 8

	Выбор объекта	Используется для выделения элемента управления, раздела или формы.
	Надпись	Создание надписи для вывода на экран поясняющего текста
	Поле	Создание поля для вывода или изменения данных
	Группа	Создание группы элементов для выбора одного значения
	Выключатель	Создание кнопки, нажатие которой возвращает значение "истина"
	Переключатель	Создание кнопки в группе элементов с единственным выбором
	Флажок	Создание индикатора для отображения логического значения
	Поле со списком	Создание поля для ввода данных или выбора значения из списка
	Список	Создание списка значений
	Рисунок	Создание рамки для вывода на экран рисунка

Предположим, мы хотим, чтобы результат использования переключателя в форме отражался в исходной таблице. Для этого предварительно нужно создать в таблице новое поле.

6. Представьте таблицу **СТУДЕНТЫ** в режиме конструктора и создайте новое поле **Общежитие** (текстовое), в столбце Описания введите текст(1 - проживает в общежитии, 2 - проживает на квартире). Поле не заполняйте данными.

Когда вы будете устанавливать переключатель формы в одно из положений (да/нет), в таблице этот результат будет отражаться цифрами (1/2).

Добавим в форму ИНФОРМАЦИЯ О СТУДЕНТАХ новый элемент управления -переключатель (для заполнения данными поля **Общежитие**).

7. Выберите инструмент "Группа".

8. Определите место в форме, куда нужно поместить верхний левый угол группы и щелкните мышью. С этого момента начинает свою работу **Мастер по разработке групп**.

В первом окне диалога **Создание группы** введите значения "проживает в общежитии" и " проживает на квартире". Нажмите кнопку **Далее**. Во втором окне мастера откажитесь от необходимости в выборе значения по умолчанию. Нажмите **Далее**. В следующем окне нажмите **Далее**, чтобы использовать значения, предложенные по умолчанию. Нажмите **Далее**. В четвертом окне щелкните **Сохранить значение в поле** и в списке полей выберите **Общежитие**, Далее. В

следующем окне определите вариант оформления группы (Вдавленное, утоплено и т.д.), выберите тип элементов группы (флажки, переключатели, выключатели). **Далее.** Далее введите подпись для группы Проживание и нажмите кнопку Готово.

Форма будет представлена в режиме конструктора. В случае необходимости переместите вновь созданный элемент управления в нужное место формы. Прокрутите все записи формы и для каждой из них установите переключатель в нужное положение.

Откройте таблицу СТУДЕНТЫ и проверьте, каким образом отражены в ней значения поля Общежитие.

Создайте автоформу: в столбец (таблица СЕССИЯ); ленточную (таблица СТУДЕНТЫ); табличную (таблица СТИПЕНДИЯ).

### Лабораторная работа №3

#### Запросы в ACCESS

С помощью запросов можно просматривать, анализировать и изменять данные из нескольких таблиц. Они также используются в качестве источника данных для форм и отчетов.

Применение запросов позволяет наиболее быстрым способом получать информацию из базы данных.

Запросы используют тогда, когда вы задаетесь вопросам и Кто, Что, Когда, Где и Как. Запросы дают вам набор данных, называемый выборкой. Выборка представляет собой результаты выполнения запроса, то есть поля из записей, удовлетворяющих поставленным вами условиям.

Рассмотрим формирование запросов на примере созданной ранее БД SESS.mdb

1. Откройте окно БД SESS.mdb и выберите вкладку ЗАПРОСЫ, затем щелкните на кнопке создать "Создать", и выберите в открывшемся окне "Простой запрос" и щелкните на кнопке ОК (рис. 9)

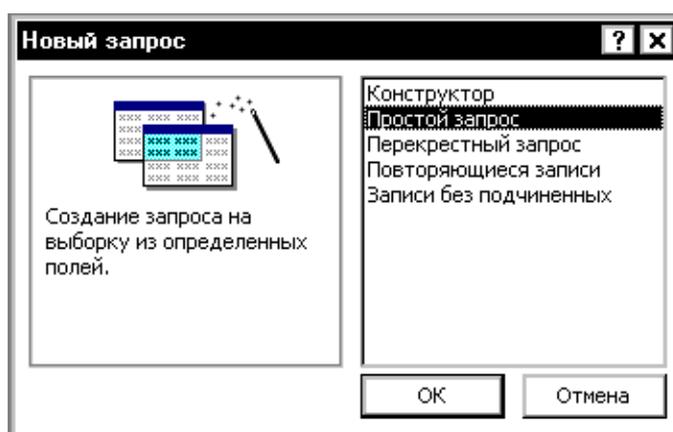


Рис. 9.

*Мастер простой запрос.*

Мастер Простой запрос дает возможность не забивать себе голову размышлениями о том, как они работают. Прежде всего необходимо выбрать определенные поля и сообщить свои пожелания по поводу представления информации. Создание запросов осуществляется по шагам:

1. В списке «Таблицы/Запросы» из таблицы выбираются поля, а затем двойным щелчком поместите в запрос нужные поля для запросов;
2. Выбираются способы представления записей: подробный подразумевает просмотр всех записей, итоговый – выбор способа обработки данных;
3. При нажатии на кнопку ИТОГИ вы увидите список заданных вами полей, содержащих числовые значения или денежную единицу – и для каждого из них параметры SUM, AVg, Min, Max (сумма, среднее, минимальное, максимальное значения) Выберите нужные параметры.
4. Спрашивается имя запроса и предлагается вывести готовые результаты, либо переключиться в режим конструктора запросов.

Сохранять запрос не надо, так как мастер делает это автоматически.

2. Создайте простой запрос «Успеваемость» для анализа результата сессии: количество «отл», «хор1», «хор», «удов».

В таблице СТИПЕНДИЯ выберите поля «Итог», «Процент», в таблице СЕССИЯ – поле «Результат».

Выберите итоговый способ обработки данных и нажмите кнопку ИТОГИ. Откроется окно изображенное на рис.2.

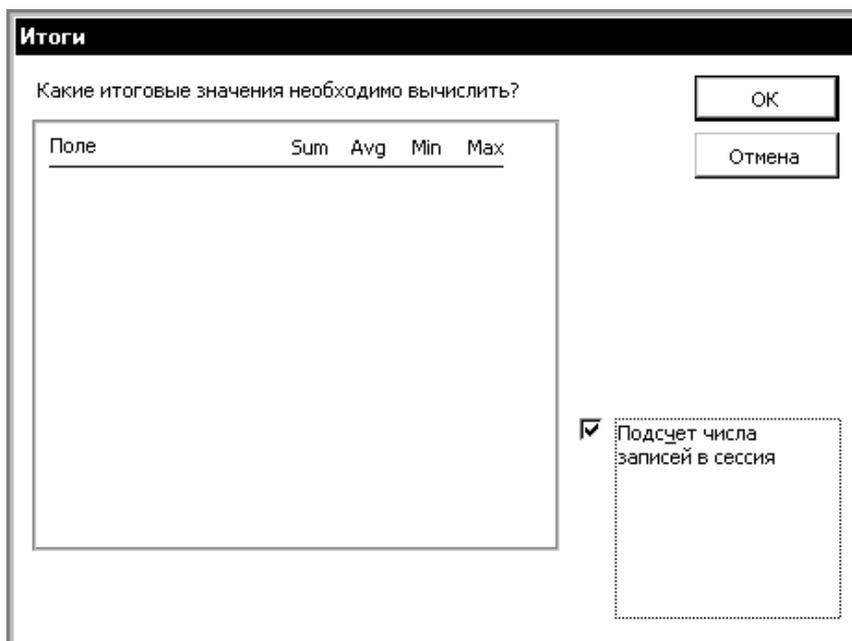


Рис.10.

Установите флажок "Подсчет записей сессия" для вычисления общего числа повторяющихся записей (Count Records In)

Выведите результаты запроса на экран.

*Конструктор запросов.*

В отличие от мастера простых запросов в конструкторе запросов можно вводить критерии отбора, как это делается в фильтрах.

1 этап - в открывшемся окне диалога "Добавление таблицы" выберите таблицы для использования в запросе, используя кнопку "Добавить".

2 этап - выбираются поля для запроса, путем перетаскивания его в первый свободный столбец в строке полей. Ширину столбцов можно увеличить при необходимости. Чтобы поместить в список все поля таблицы, дважды щелкните на имени таблицы.

3 этап - щелкните мышью в строке "Условия отбора" в столбце требуемого поля и введите подходящее значение. Для этого используются операторы:

<, >, <=, >=, <>

is null, is not null - для выявления пустых и непустых строк

between... and - для проверки вхождения в интервал значений

and, or - для связки условий.

4 этап - обратите внимание на строку "Вывод на экран". При установленном флажке в соответствующем столбце поле будет присутствовать в выборке, создаваемой запросом.

Из окна конструктора можно выполнить запрос с помощью кнопок "Запуск\*" и «Представление запроса» (переключение в режим таблицы).

Для добавления в запрос новой таблицы открывают окна базы данных, выбирают нужный запрос на вкладке «Запросы» и щелкают на кнопке «Конструктор». В окне «Добавление таблицы» добавьте нужную таблицу для дополнения запроса (смотрите 1-4 этап) Если между таблицами установлены связи, то Access покажет их. При наличии нескольких связанных таблиц Access объединяет таблицы по связанным полям. Чтобы избежать некоторых проблем, возникающих при объединении таблиц, можно изменить тип их объединения. Дважды щелкните на линии связи, и на экране появится окно диалога.

3. Создайте запрос Студенты – ИТОГ для вывода итогов сдачи сессии каждого студента. Для этого откройте конструктор запросов и добавьте в него таблицы СТУДЕНТЫ, СЕССИЯ и СТИПЕНДИЯ. Выберите поля для запроса: ФИО, ГРУППА, ПОЛ, РЕЗУЛЬТАТ. Просмотрите запрос.

4. Запишите следующие условия отбора записей:

а) студентов, получающих стипендию,

б) сортировка фамилий студентов по возрастанию,

в) студентов женского пола.

#### **Универсальные запросы с параметром.**

Для постоянного изменения какого-либо условия в запросе можно в отбора поля удалить текст, а затем либо непосредственно, либо в окне «Область ввода» (клавиши SHIFT+F2) задать подсказку в виде [Введите...]. Параметр служит подсказкой, благодаря которой при запуске запроса вы узнаете, чего от вас хотят. Для большей гибкости при отборе записей можно задавать не один, а несколько параметров.

Создайте универсальный запрос Студенты-ИТОГ. В поле «Группа» установите область ввода [введите номер группы], в поле «Результат» - область ввода [В ведите процент стипендии 200/150/100/0].

Вычисление по полям

Чтобы произвести вычисление, запрос открывают в режиме КОНСТРУКТОРА и в строке ПОЛЕ вводится математическое выражение. Например, для вычисления среднего балла экзаменационных оценок запишем:

сумма: ([сессия]![оценка1]+[оценка2]+ [оценка2])/3, где *сумма*- название будущего столбца.

[сессия]! [оценка1]–имя таблицы, заключенная в квадратные скобки, восклицательный знак, затем имя поля. Для вычисления используются математические операторы.

6. Создайте новый запрос СУММА, содержащий все поля запроса Студенты -ИТОГ. Запрос СУММА откройте в режиме конструктора и добавьте поля: Размер (для указания минимальной стипендии в рублях), Сумма (для вычисления размера стипендии в рублях).

*Перекрестные запросы.*

Запросы этого рода предназначены для анализа информации, основанного на двух и более полях из одной или более таблиц. К сожалению, мастер перекрестных запросов использует поля только из одной таблицы или запроса Выберите мастера перекрестного запроса.

1. окно -укажите таблицу или запрос, по которой вы хотите построить перекрестный запрос. Если в списке перечислены таблицы, то выберите в группе «Показать» позицию «Запросы». Выделите нужную таблицу.

2 окно - укажите поле, которое будет использоваться для заголовков строк,

3 окно -запрашивается поле для заголовков столбцов.

4 окно - указывается поле, по которому производятся вычисления, и характер вычислений.

На данном этапе необходимо обратить внимание на установку флажка «Да», «Нет», расположенный под текстом «Вычислить итоговое значение для каждой строки».

5 окно - придумайте запросу имя и сообщите о ваших дальнейших действиях.

7. Создадим перекрестный запрос для анализа успеваемости по группам: число отличников, хорошистов и т.д. в группах, в виде:

Итог	Группа1	Группа2	Группа3	Группа4
Отл				
Хор1				
Хор				
Удовл				

Выберем запрос Студенты-ИТОГ в качестве основного. В качестве заголовка строк будем использовать поле ИТОГ, для заголовка столбцов - поде ГРУППА. На пересечении строки и

столбца мы должны получать число студентов данной группы, имеющих заданный итог, поэтому выберем подсчет числа фамилий повторяющихся записей – Count (Фамилия).

### **Фильтры и запросы.**

Любой запрос или фильтр можно преобразовать в фильтр или наоборот. Когда на экране находится окно фильтра, в меню «Файл» можно выбрать «Сохранить как запрос» или «Загрузить из запроса».

Создайте фильтр для поиска студентов, живущих в г. Зея, для этого загрузите фильтр из запроса СТУДЕНТЫ -ИТОГ.

Завершите работу с Access.

## **MS PowerPoint**

Приложение MS PowerPoint предназначено для создания и редактирования произвольных презентаций.

Презентация (от английского presentation - представление) - это передача, представление аудитории новых для нее идей, планов, разработок. Компьютерная презентация - это файл, в который такие материалы собраны. Файл презентации состоит из последовательности кадров, или слайдов. Эти слайды можно не только выводить на экран компьютера или специального проектора по ходу выступления, но и напечатать на бумаге или прозрачной пленке.

### ***Создание новой презентации***

При запуске PowerPoint появляется диалоговое окно, которое имеет четыре переключателя, позволяющих:

1. Создать новую презентацию, используя мастер автосодержания, шаблон презентации, пустую презентацию.
2. Открыть имеющуюся презентацию.

Мастер автосодержания помогает выбрать один из нескольких шаблонов содержания и способы его оформления. Шаблон представляет собой презентацию с заранее установленными цветами слайда и текстовыми стилями.

После выбора переключателя пустая презентация также открывается диалоговое окно с 24 макетами разметки слайдов. В новой презентации используются цветовая схема, стиль заголовка и стили текста презентации, принимаемые по умолчанию.

Переключатель открыть презентацию позволяет открыть существующую презентацию PowerPoint.

Если PowerPoint уже запущен, то можно создать новую презентацию:

1. используя мастер автосодержания командой Файл - Создать, выбрать вкладку Общие, затем Мастер автосодержания
2. по шаблону командой Файл - Создать, выбрать вкладку Презентации или Шаблоны оформления
3. без использования шаблона командой Файл - Создать, выбрать вкладку Общие, затем Новая презентация или использовать кнопку Создать на Стандартной панели инструментов.

### ***Создание слайдов по разметке***

При создании нового слайда можно выбрать один из макетов. Макет предусматривает, что в каждой зоне слайда будет располагаться определенный вид информации. Существует макет, включающий рамки для заголовка, текста и диаграммы; в другом макете предусмотрены места для заголовка и картинки. Расположение, размер и формат рамок допускается изменять. Чтобы изменить макет уже созданного слайда, нужно воспользоваться командой Формат - Разметка слайда или нажать кнопку Разметка слайда на панели инструментов Стандартная, затем выбрать новую разметку.

### ***Режимы PowerPoint***

Режимами называются способы отображения и работы над презентацией. Кнопки в нижней части окна презентации (рисунок 4) позволяют переходить от одного вида режима к другому. Также все эти режимы доступны из меню Вид.

В обычном режиме отображаются три области: область структуры, область слайда и область заметок. Эти области позволяют одновременно работать над всеми аспектами презентации. Размеры областей можно изменять, перетаскивая их границы.

Область структуры служит для организации и развертывания содержимого презентации. В ней можно вводить текст презентации и упорядочивать пункты списка, абзацы и слайды.

В области слайда отображается текст каждого слайда с учетом форматирования. На отдельные слайды можно добавлять рисунки, фильмы, звуки, анимацию и гиперссылки.

Область заметок служит для добавления заметок докладчика или сведений для аудитории.

В режиме структуры презентация изображается в сокращенном виде: только заголовки слайдов и содержащийся в них текст. В верхнем правом углу находится окно для просмотра текущего слайда.

Работа со структурой является наилучшим вариантом организации и развития презентации, так как в процессе работы на экране отображаются все заголовки и основной текст. Можно поменять местами пункты на слайде, переместить слайд целиком с одной позиции в другую и изменить заголовки и основной текст.

В режиме слайдов создают и редактируют отдельные элементы слайдов: текст в метках-заполнителях, объекты (графику, звук и т.п.), дополнительные тексты, фигуры, собственные

рисунки и т.д. С одного слайда на другой переключаются при помощи вертикальной полосы прокрутки, а также клавишами Page Up и Page Down.

В режиме сортировщика слайдов на экране в миниатюре отображаются сразу все слайды презентации в том порядке, в каком их будут показывать во время выступления. Это упрощает добавление, удаление (клавиша Delete), копирование (используя клавишу Ctrl) и перемещение слайдов, задание времени показа слайдов и выбор способов их смены.

В режиме показа слайдов представляют презентацию во время выступления. Чтобы выбрать какой-либо способ показа, нужно в Режиме слайдов выполнить команду Показ слайдов - Настройка презентации и установить в соответствующее положение переключатель. Существует три разных способа показа слайдов:

- управляемый докладчиком (полный экран). Это обычный способ проведения показа, управляемого докладчиком, когда слайды отображаются в полноэкранном режиме. Ведущий получает полный контроль над презентацией. Он может проводить ее вручную (по щелчку мыши) или автоматически (через определенное время);
- управляемый пользователем (окно). В этом случае показ слайдов будет проводиться в стандартном окне со специальными меню и командами, обеспечивающими возможность самостоятельного просмотра презентации пользователем;
- автоматический (полный экран). В этом режиме презентация будет проводиться в автоматическом режиме, который возобновляется после 5 минут простоя. Это можно использовать на выставочном стенде или собрании. Зрители могут менять слайды, переходить по гиперссылкам, но не могут изменять презентацию. Прервать показ слайдов можно, нажав клавишу Esc.

### ***Вставка слайдов в PowerPoint***

Вставить слайды можно следующими способами:

1. При помощи команды Вставка -Новый слайд или с помощью кнопки Новый слайд на Стандартной панели инструментов.

Чтобы добавить новый слайд, который имеет ту же разметку, что и текущий слайд, нужно щелкнуть на кнопке Новый слайд на Стандартной панели инструментов, удерживая клавишу Shift.

2. В режиме структуры. Для создания «чистого» слайда нужно поместить курсор в конец предыдущего слайда и нажать клавишу Enter. Если прототип слайда уже где-то существует (в другой презентации или в текстовом документе), то можно вставить его в нужное место презентации, используя буфер обмена.

3. При помощи команды Вставка - Дублировать слайд. Для этого необходимо выделить уже имеющийся слайд презентации и выбрать эту команду. Копия слайда будет расположена сразу после его прототипа.

4. При помощи команды Вставка - Слайды из файлов. Эта команда позволяет копировать из одной презентации в другую.

5. При помощи команды Вставка - Слайды из структуры. Под структурой в данном случае подразумевается любой текстовый файл, абзацы которого PowerPoint преобразует в слайды презентации.

### ***Изменение цвета фона и способа заливки слайда***

Для выбора цвета, текстуры, узора или рисунка для фона слайда используется команда, Формат - Фон. Появляется диалоговое окно Фон. В группе Заливка фона нужно щелкнуть по кнопке списка. Для изменения цвета, входящего в цветовую схему, необходимо выбрать один из восьми цветов, а чтобы изменить цвет, не входящий в цветовую схему, нажать кнопку Другие цвета. На вкладке Обычные выбирается нужный цвет, на вкладке Спектр можно создать свой цвет.

Для изменения заливки в диалоговом окне Фон нужно щелкнуть по кнопке списка, затем выбрать Способы заливки, перейти на одну из вкладок Градиентная, Текстура, Узор или Рисунок.

Для восстановления цвета фона и заливки, используемого по умолчанию, используется кнопка Авто.

Применение внесенных изменений к текущему слайду выполняется щелчком по кнопке Применить, а ко всем слайдам - по кнопке Применить ко всем.

### ***Ввод текста***

Самый простой способ добавления текста к слайду - ввести его непосредственно в рамку на слайде. Чтобы вставить текст вне рамки, можно воспользоваться инструментом Надпись, расположенным на панели инструментов Рисование или командой Вставка - Надпись.

Чтобы добавить текст в автофигуру, необходимо щелкнуть в ней и набрать текст или с помощью инструмента Надпись.

Для изменения внешнего вида шрифта используется команда Формат - Шрифт или кнопки на панели инструментов Форматирования.

### ***Способы вставки рисунков в презентацию***

В комплект PowerPoint входит стандартный набор рисунков в виде коллекции.

Для выбора рисунка нужно выполнить команду Вставка - Рисунок или нажать кнопку Добавить картинку на Стандартной панели инструментов и перейти на вкладку Графика или Картинки.

Для вставки рисунка необходимо сначала выбрать нужный раздел, затем подходящий рисунок и нажать кнопку Вставить или выполнить двойной щелчок по рисунку.

Чтобы вставить рисунок из другого приложения, нужно выполнить команду Вставка - Рисунок, затем выбрать Из файла. При этом появится панель инструментов Настройка изображения, с помощью которой можно перекрасить рисунок, обвести его рамкой, отрегулировать яркость и контрастность.

Чтобы вставить сканированную фотографию, нужно выполнить команду Вставка - Рисунок, затем выбрать Со сканера. Изображение откроется в приложении Microsoft Photo Editor, где его можно изменить.

### ***Анимация и способы перехода слайдов***

Анимацией называется звуковое и визуальное оформление, которое можно добавить к тексту или другому объекту, например, к диаграмме или рисунку. Переходом называется специальный эффект, используемый для отображения очередного слайда в ходе показа презентации. (Например, можно обеспечить появление слайда из затемнения или наплывом.) Анимация, способы смены слайдов и другие средства используются для подчеркивания определенных аспектов сообщаемых сведений, а также делает презентацию более привлекательной. Для каждого объекта слайда также можно установить способ его появления на экране, (например, «влетание» с левой или с правой стороны, потускнение или изменение в цвете).

### ***Добавление переходов и условий переходов в показ слайдов***

1. В Режиме слайдов или Режиме сортировщика необходимо выделить слайды, для которых следует добавить переходы.

2. В меню Показ слайдов выбрать команду Переход слайда.

3. В появившемся диалоговом окне Переход слайда в списке Эффект выбрать нужный вариант эффекта перехода, скорость показа эффекта (медленно, средне, быстро) и условия перехода (по щелчку мыши или автоматически). (Если установлены оба флажка По щелчку мыши и Автоматически, то следующий слайд появляется по щелчку мыши или автоматически по истечении интервала, заданного в поле секунд (в зависимости от того, что произойдет раньше). Если сброшены оба флажка, то следующий слайд появляется только после выбора в контекстном меню команды Далее.)

4. Чтобы применить переход к выделенному слайду, нажать кнопку Применить. Чтобы применить переход ко всем слайдам, нажать кнопку Применить ко всем.

### ***Выполнение анимации текста и объектов***

Самый быстрый способ создания простейшей анимации: выделить нужный объект (в Режиме слайдов), выполнить команду Показ слайдов - Встроенная анимация и выбрать подходящий вариант.

Чтобы предварительно просмотреть анимацию в Режиме слайдов, необходимо выполнить команду Показ слайдов - Просмотр анимации. Анимация будет показана в миниатюрном слайде. Чтобы снова начать показ, нужно щелкнуть эту миниатюру слайда.

### ***Запуск показа слайдов из PowerPoint***

Для запуска презентации нужно выполнить любое из приведенных ниже действий:

1. Перейти в режим Показ слайдов.
2. Выполнить команду Показ слайдов - Показ.

3. Выполнить команду Вид - Показ слайдов.

### ***Форматы сохраненных файлов***

Презентацию (новую или сохранявшуюся ранее) можно сохранить в ходе работы над ней командой Файл - Сохранить или использовать кнопку Сохранить на панели инструментов Стандартная. Также можно сохранить копию презентации под другим именем или в другом месте командой Файл - Сохранить как. Можно выбрать различные форматы сохраняемых файлов:

1. расширение файла обычной презентации ppt;
2. презентация, которая всегда будет открываться в режиме Показ слайдов, имеет расширение pps.

**Задание:** создайте презентацию (10 слайдов) на выбранную вами тему.

## **Лабораторные работы по теме язык HTML**

### **Тема 1: Введение в язык HTML**

#### **Создание HTML-документов.**

HTML-документы могут быть созданы при помощи любых текстовых редакторов или специализированных HTML-редакторов и конвертеров.

HTML-документ — это обычный файл в формате ASCII. В его основе лежат специальные дескрипторы (теги), которые и определяют правила форматирования данных. Код программы на языке HTML — обычный текстовый файл, поэтому проще всего написать его в Блокноте — стандартном Windows-приложении, которое вызывается командой Пуск → Программы → Стандартные → Блокнот.

Когда программа написана, ее файл следует сохранить с расширением .htm. Имя HTML-файла следует писать строчными латинскими буквами без пробелов. Можно использовать цифры, знаки подчеркивания и тире. Как только файл будет записан с расширением .htm, его значок сразу примет вид «интернетовского». Файл с расширением .htm можно просмотреть в программе Internet Explorer.

HTML — это язык тегов. Под тегами понимаются специальные управляющие коды, записываемые в тексте в угловых скобках. Теги, окружающие текстовые или графические команды, определяют параметры форматирования текста, местоположение рисунков относительно текста, расположение объектов на странице.

Все HTML-теги начинаются с символа левой угловой скобки (<) и заканчиваются символом правой угловой скобки (>).

Почти все теги образуют пары, то есть для открывающего тега существует закрывающий тег. Например: открывающий и закрывающий теги, определяющие заголовок документа: <TITLE> Заголовок документа </TITLE>. Есть и непарные теги, например <BR>, <AREA>, но их мало.

Закрывающий тег выглядит так же, как открывающий, и отличается от него символом прямого слэша перед текстом внутри угловых скобок. В данном примере тег `<TITLE>` говорит веб-браузеру об использовании формата заголовка, а тег `</TITLE>` — о завершении текста заголовка.

Некоторые теги, такие как `<P>` (тег, определяющий абзац), не требуют завершающего тега, но его наличие придает исходному тексту документа стройность, делает его понятнее.

Язык HTML нечувствителен к регистру символов. Дополнительные пробелы, символы табуляции и конца строки, добавленные в исходный текст HTML-документа, чтобы сделать его понятнее, игнорируются веб-браузером при интерпретации документа. HTML-документ может включать вышеописанные элементы, только если они помещены внутрь тегов `<PRE>` и `</PRE>`. Вся программа состоит из набора тегов. Теги могут иметь параметры, или атрибуты, которые играют роль дополнения тегов.

### Структура документа

Когда веб-браузер получает документ, он по тегам определяет, как документ должен быть интерпретирован.

Тег `<HTML>`. Данный тег сообщает веб-браузеру, что документ написан на языке HTML.

Теги `<HTML>` и `</HTML>` образуют для HTML-документа так называемый контейнер.

Тег `<HEAD>`. Структурно документ распадается на две части: заголовочную и основную, или тело документа. Заголовочная часть размещается между тегами `<HEAD>` и `</HEAD>`, основная — между тегами `<BODY>` и `</BODY>`.

Тег `<TITLE>`. Внутри контейнера `<HEAD>` и `</HEAD>` размещается единственный обязательный контейнер из тегов `<TITLE>` и `</TITLE>`, содержащий текст заголовка.

Большинство веб-браузеров отображают содержимое тега `<TITLE>` в заголовке окна, содержащего документ, и в файле закладок, если он поддерживается. Заголовок документа при отображении самого документа в окне браузера не виден.

Теги тела документа идентифицируют отображаемые в окне компоненты HTML-документа. Тело документа может содержать ссылки на другие документы, текст и другую форматированную информацию

Тег `<BODY>`. Тело документа должно находиться между тегами `<BODY>` и `</BODY>`. Это та часть документа, которая отображает его текстовую и графическую информацию.

Тег `<BODY>` парный. Между открывающим тегом `<BODY>` и закрывающим тегом `</BODY>` размещаются все другие теги программы, составляющие основное содержание документа. Тег `<BODY>` предназначается для выделения той части документа, которая должна быть показана пользователю на экране.

Пример создания веб-страницы с текстом:

```
<HTML> <HEAD> <TITLE> Моя первая веб-страница </TITLE> </HEAD>
```

```
<BODY>
```

33 удивительнейшие буквы русского алфавита передают человечеству события, факты, настроения, переживания, радость и вдохновение, то есть все, что можно видеть, слышать, пробовать, трогать, чем можно наслаждаться. Как же буквы могут передавать огромное количество информации? Буквы образуют слово, слова собираются в предложения, предложения могут составлять целую книгу. Книги состоят из текста и рисунков.

```
</BODY> </HTML>
```

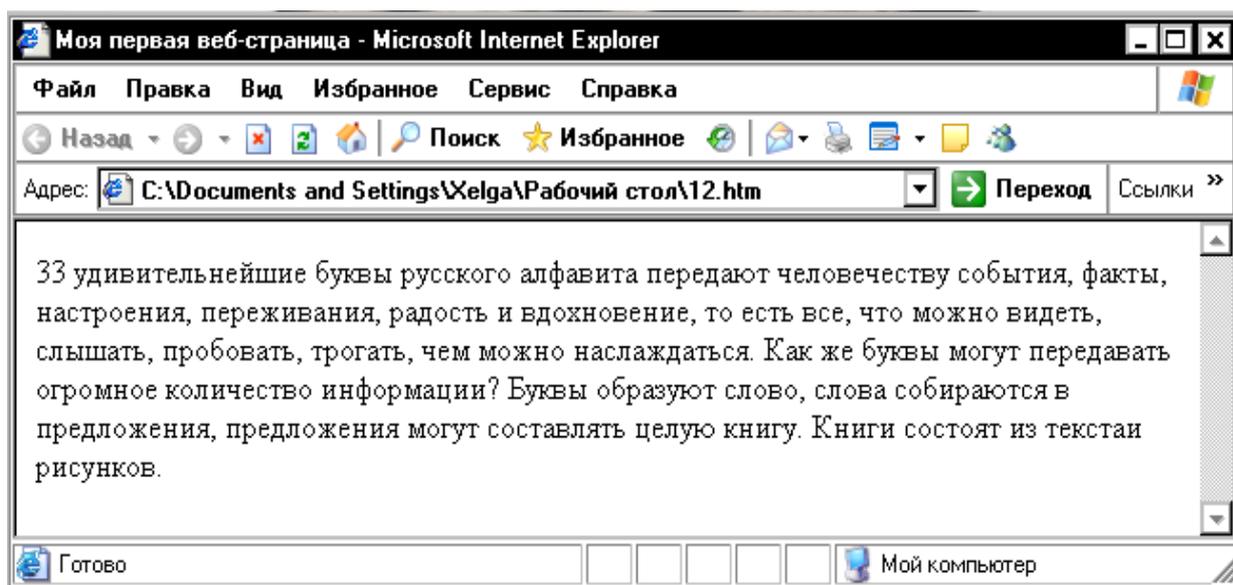


Рис. 1. Веб-страница с текстом.

### Атрибуты

Атрибуты — дополнительные управляющие слова, отделенные от тега и друг от друга пробелами. Атрибуты — это имена свойств тегов, которые могут принимать определенные

значения. Атрибуты имеются только у открывающих тегов, у закрывающих тегов их нет. Атрибуты влияют на результат интерпретации тега браузером.

Некоторые атрибуты имеют конкретное значение, которое задается после знака равенства.

### Атрибуты тега <BODY>

Начальный тег <BODY> может иметь несколько атрибутов

Атрибут BACKGROUND задает графическое изображение, которое заполнит фон документа. Файл с изображением должен быть сохранен в формате GIF или JPEG. Пример:

```
<BODY BACKGROUND="(URL)(путь) имя файла">
```

Файл с изображением фона лучше размещать в том же каталоге, что и файл самого документа, тогда URL-адрес и путь указывать не нужно.

Атрибут BGCOLOR задает цвет фона документа при помощи шестнадцатеричных значений интенсивности цветовой модели RGB (Red, Green, Blue — красный, зеленый, синий) или строчного литерала, соответствующего названию цвета.

Атрибут TEXT задает используемый по умолчанию цвет текста, не являющегося гиперссылкой. По умолчанию такой текст черный. Пример:

```
<BODY TEXT="цвет">.
```

Атрибут LINK задает цвет гиперссылки; в большинстве браузеров он по умолчанию темно-синий. Пример:

```
<BODY LINK ="цвет">.
```

Атрибут ALINK задает цвет активной гиперссылки, который меняется в момент щелчка на ней мышью и который желательно делать отличным от цвета фона (задаваемого атрибутом LINK).

Пример:

```
<BODY ALINK ="цвет">.
```

Атрибут VLINK задает цвет посещенной гиперссылки, который желательно делать отличным от цвета фона (задаваемого атрибутом LINK) и от цвета активной гиперссылки (задаваемого атрибутом ALINK). Пример:

```
<BODY ALINK ="цвет">.
```

Атрибут BGPROPERTIES задает свойства фонового изображения. В данный момент браузерами поддерживается единственное его значение fixed, запрещающее прокрутку изображения. Пример:

```
<BODY BGPROPERTIES="fixed">.
```

Атрибут TOPMARGIN задает верхнюю границу страницы в пикселах. Пример:

```
<BODY TOPMARGIN=число>.
```

Атрибут BOTTOMMARGIN задает нижнюю границу страницы в пикселах. Пример:

```
<BODY BOTTOMMARGIN =число>.
```

Атрибут LEFTMARGIN задает левую границу страницы в пикселах. Пример:

<BODY LEFTMARGIN=число>

Атрибут RIGHTMARGIN задает правую границу страницы в пикселах. Пример:

<BODY RIGHTMARGIN=число>

Пример создания веб-страницы, для которой заданы границы:

```
<HTML> <HEAD> <TITLE> Моя первая программа </TITLE> </HEAD>
<BODY bottomMargin=150 bgColor=#FFD000 leftMargin=150 rightMargin=150>
<H2> А. С. Пушкин </H2>
<H2> Зимнее утро </H2>
<BR> Мороз и солнце; день чудесный;
<BR> Еще ты дремлешь друг прелестный,
<BR> Пора, красавица, проснись:
<BR> Открой сомкнуты негой взоры
<BR> Навстречу северной Авроры,
<BR> Звездой Севера явись!
</BODY> </HTML>
```

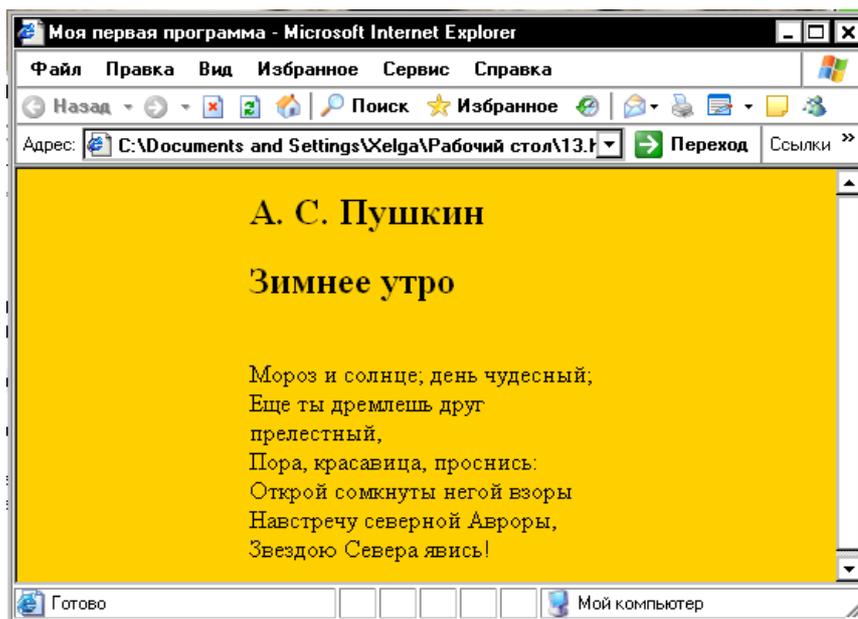


Рис. 2. Веб-страница, для которой заданы границы.

### Уровни заголовков

Заголовки различных уровней позволяют структурно разделить формально не ограниченный по длине текст на отдельные разделы и абзацы. Первый уровень заголовков (самый высокий) обозначается цифрой 1, следующий — 2, и т. д. Теги с меньшими номерами определяют заголовки более высоких уровней. Большинство браузеров поддерживает интерпретацию шести уровней заголовков, определяя каждому из них собственный стиль. Заголовок самого верхнего уровня имеет признак 1. Синтаксис заголовка первого уровня:

<H1> Заголовок первого уровня </H1>

Заголовок любого уровня может быть представлен в общем случае так:

`<Hx>` Заголовок x-го уровня `</Hx>` Здесь x — цифра от 1 до 6, определяющая уровень заголовка.

Атрибут `ALIGN` управляет горизонтальным выравниванием, принимая значение `LEFT` (влево), `CENTER` (по центру), `RIGHT` (вправо) или `JUSTIFY` (по ширине от левой границы до правой). Например: `<ALIGN=CENTER>`

Пример создания веб-страницы с разными заголовками:

```
<HTML> <HEAD> <TITLE> Заголовки </TITLE> </HEAD>
```

```
<BODY>
```

```
<H1> Заголовок 1 </H1>
```

```
<H2> Заголовок 2 </H2>
```

```
<H3> Заголовок 3 </H3>
```

```
<H4> Заголовок 4 </H4>
```

```
<H5> Заголовок 5 </H5>
```

```
<H6> Заголовок 6 </H6>
```

```
</BODY> </HTML>
```

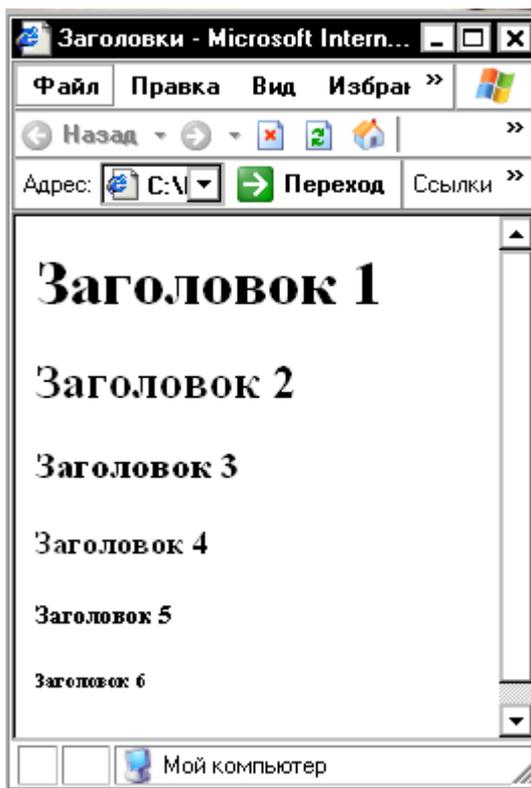


Рис. 6. Веб-страница с разными заголовками.

## Линии

Горизонтальные линии задаются тегом `<HR>`. Это одинарный тег, закрывающего тега он не имеет. По умолчанию линия является объемной и с тенью. Горизонтальные линии позволяют разделить длинный формально не ограниченный текст на отдельные разделы.

Атрибут `<ALIGN>` задает способ выравнивания линейки:

`LEFT` – слева,

`RIGHT` – справа,

`CENTER` - по центру.

Атрибут `WIDTH` определяет ширину (длину) линейки относительно страницы. Его значение может быть задано в пикселях или в процентах, например: `<HR WIDTH="80%">`. Реальную толщину определяет атрибут `SIZE`, его значение задается в пикселях. Чтобы линейка

отображалась в одноцветном режиме необходимо указать атрибут NOSHADE, не требующий указания значения, например: <HR WIDTH="100%" ALIGN="RIGHT" SIZE="5" NOSHADE>.

Синтаксис:

```
<HR SIZE=number WIDTH= number | percent ALIGN=left | right | center NOSHADE>
```

Пример веб-страницы с линиями разной толщины и длины:

```
<HTML> <HEAD> <TITLE> Линии </TITLE> </HEAD>
<BODY>
<HR>
<HR ALIGN=CENTER WIDTH="20%" noshade size=40>
<HR ALIGN=CENTER WIDTH="30%" noshade size=30>
<HR ALIGN=CENTER WIDTH="40%" noshade size=20>
<HR ALIGN=CENTER WIDTH="50%" noshade size=10>
<HR ALIGN=CENTER WIDTH="60%" noshade size=5>
</BODY>
</HTML>
```

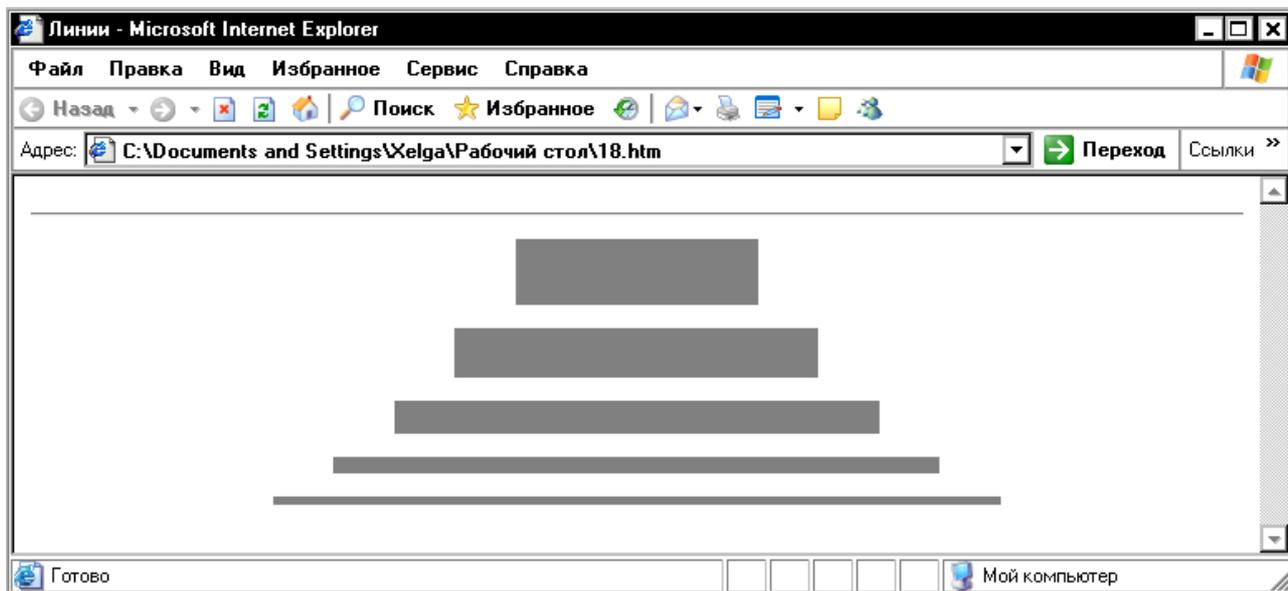


Рис. 7. Веб-страница с линиями разной толщины и длины.

Пример веб-страницы с объемными линиями выровненными по правому краю:

```
<HTML> <HEAD> <TITLE> Линии, выровненные по правому краю </TITLE>
</HEAD>
<BODY BGCOLOR="#00FF00">
<HR ALIGN=LEFT WIDTH="35%" size=5>
<HR ALIGN=LEFT WIDTH="45%" size=10>
```

```

<HR ALIGN=LEFT WIDTH="55%" size=15>
<HR ALIGN=LEFT WIDTH="65%" size=20>
</BODY>
</HTML>

```

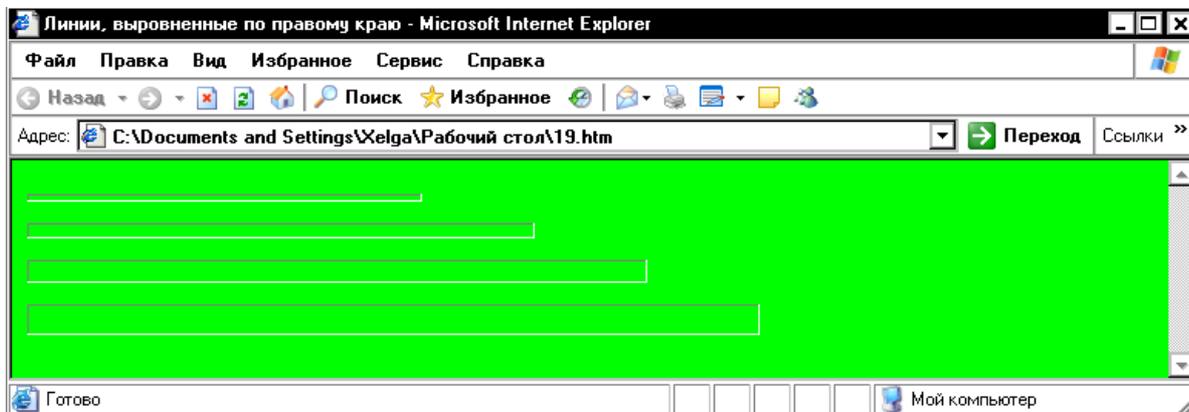


Рис. 8. Веб-страница с объемными линиями выровненными по правому краю.

### Абзацы, выравнивание

В HTML документе невозможно разбить документ на абзацы, используя клавишу Enter. Отдельный абзац задается с помощью тега <P>. По умолчанию текст, графика и элементы таблицы выровнены по правому краю. Выравнивание можно задать с помощью атрибута ALIGN. Этот атрибут может принимать значения: LEFT, RIGHT, CENTER, JUSTIFY, что позволяет выровнять абзац по левому краю, по центру, по правому краю, по ширине.

Например, следующая запись позволяет выровнять абзац по центру:

```
<P ALIGN=CENTER >
```

Сплошной текст можно разбить при помощи тега перехода на новую строку <BR>.

Пример веб-страницы с текстом, выровненным различными способами:

```
<HTML> <HEAD> <TITLE> F,pfws b hfphsds cnhjr </TITLE> </HEAD>
```

```
<P ALIGN=CENTER>
```

Язык HTML является стандартным языком, предназначенным для создания гипертекстовых документов в среде WWW. </P>

<BR> HTML-документы (или веб-документы) могут просматриваться веб-браузерами различных типов.

```
<P ALIGN=RIGHT>
```

Основное преимущество HTML заключается в том, что документ может быть просмотрен на веб-браузерах различных типов и на различных платформах. HTML - один из наиболее простых языков создания веб-страниц. </P>

```
<P ALIGN=LEFT>
```

WWW - это сообщество веб-серверов (сетевых компьютеров), входящих в Интернет, на которых хранятся текстовые, графические, видео-, аудио- и другие информационные файлы.

</P> </BODY>

</HTML>

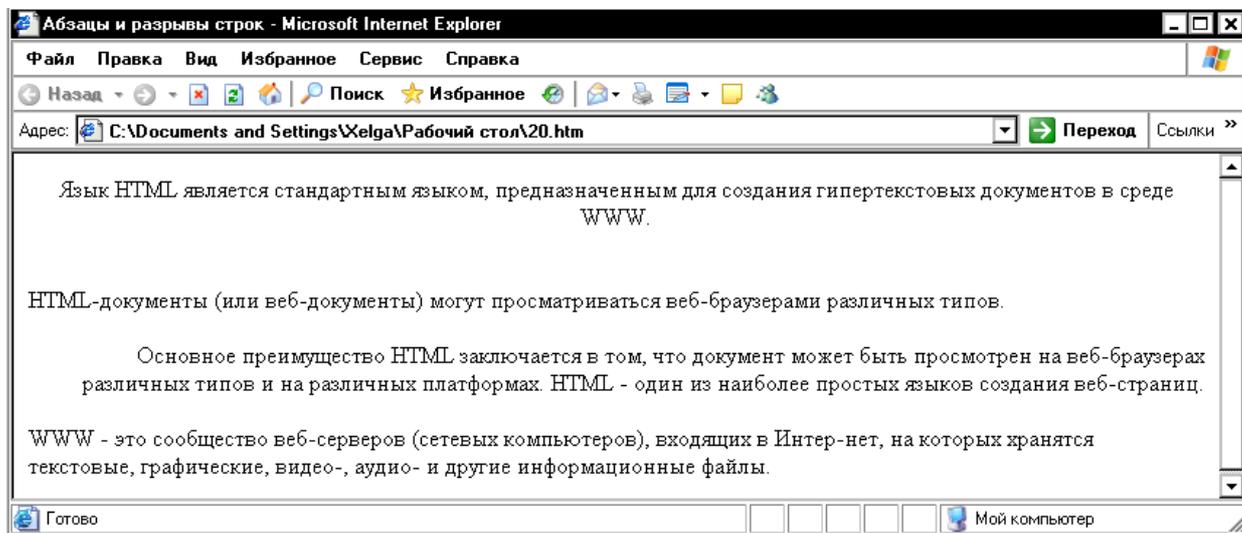


Рис. 9. Веб-страница с текстом, выровненным различными способами.

### Адресные данные

Один из самых важных элементов в документе HTML – это элемент ADDRESS. В нем вы называете себя как автора документа и сообщаете как можно связаться с вами. Здесь также можно поместить информацию об авторских правах на материал, содержащийся на странице, дату создания документа и дату последней редакции. Элемент ADDRESS обычно помещается в начале или в конце документа.

Элемент ADDRESS состоит из текста, расположенного между открывающим и закрывающим тегами <ADDRESS> </ADDRESS>. Текст внутри элемента ADDRESS отображается курсивом.

### Задания для выполнения

1. Создайте первую веб-страницу, она должна содержать:

1. заголовок окна «Моя первая веб-страница»;
2. заголовок документа: «Программное обеспечение ПК»
3. заголовок документа отделите от текста линией;
4. текст:

Программное обеспечение - совокупность программ обработки данных.

Программа - упорядоченная последовательность команд (инструкций) компьютера для решения задачи.

Программный продукт - комплекс взаимосвязанных программ для решения определенной задачи массового спроса, подготовленный к реализации как любой вид промышленной продукции.

Интегрированное программное обеспечение — набор нескольких программных продуктов, функционально дополняющих друг друга, поддерживающих единые информационные технологии, реализованные на общей вычислительной и операционной платформе.

5. текст должен быть разбит на абзацы;
6. задайте цвет фона документа;
2. В конце текста введите свои адресные данные.
3. Сохраните файл с именем start.htm в свою папку.
4. Просмотрите страницу в обозревателе Internet Explorer.
5. Выполненную работу покажите преподавателю.

## Тема 2: Форматирование текста, списки

### Шрифт

Тег <FONT> позволяет с помощью атрибутов управлять размером и цветом шрифта.

Атрибут SIZE управляет размером шрифта. Шрифт может иметь размер от 1 до 7. Можно указать размер шрифта непосредственно цифрой или задать смещение относительно базового значения (по умолчанию — 3) в положительную или отрицательную сторону. Соответствие размера шрифта величине в пунктах иллюстрирует табл. 1.

Таблица 1. Типичные размеры шрифтов и их величины в пунктах

Размер шрифта	Типичная величина в пунктах
1	8
2	10
3	12
4	14
5	18
6	24
7	36

Размер шрифта можно изменить при помощи следующей команды:

```
<FONT SIZE=+|- n>
```

Для изменения базового значения служит команда:

```
<BASEFONT SIZE=n> .
```

Пример использования шрифтов разного размера:

```
<HTML> <HEAD> <TITLE> Размер шрифта </TITLE> </HEAD>
```

```
<BODY>
```

```
<P> Б
```

```
<FONT SIZE=+1> Л
```

```
<FONT SIZE=+2> А
```

```
<FONT SIZE=+3> Г
```

```
<FONT SIZE=+4> О
```

```

<FONT SIZE=+5>В
<FONT SIZE=+5>Е
<FONT SIZE=+4>Щ
<FONT SIZE=+3>Е
<FONT SIZE=+2>Н
<FONT SIZE=+1>С
К </P>
</BODY> </HTML>

```

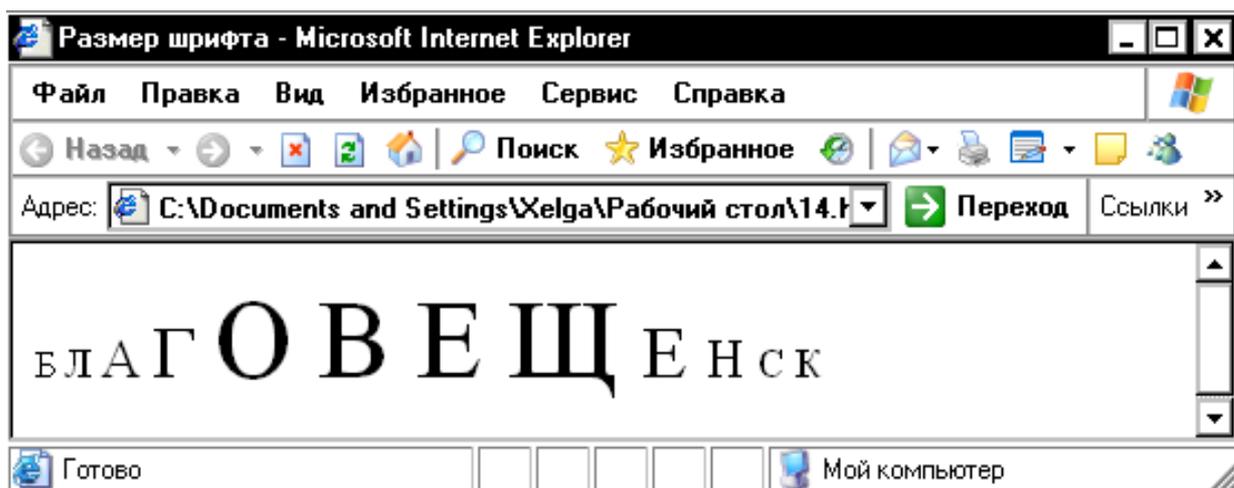


Рис. 3. Веб-страница с буквами разного размера.

Чтобы изменить цвет шрифта, используется атрибут COLOR тега FONT:

```
<FONT COLOR="#xxxxxx">
```

Цвет указывается в цветовой модели RGB соответствующими значениями цветовых составляющих в шестнадцатеричном формате. Например, белый цвет обозначается FFFFFFF, черный – 000000, синий – 0000FF и т.п.

Пример создания веб-страницы с разноцветным текстом:

```

<HTML> <HEAD> <TITLE> Цвет шрифта </TITLE> </HEAD>
<BODY>
<FONT COLOR="#FF0000">
Красный </FONT>
<FONT COLOR="#00FF00">
Зеленый </FONT>
<FONT COLOR="#0000FF">
Синий </FONT>
</BODY> </HTML>

```

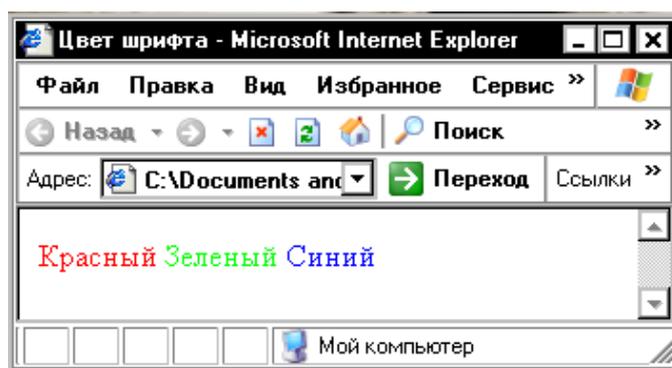


Рис. 4. Веб-страница с разноцветным текстом.

### Задание начертания шрифта

Для выделения текстовой информации в документах можно использовать различное начертание. Большинство браузеров поддерживает жирное и курсивное начертание, большой и маленький шрифты, а также подстрочный и надстрочный индексы. Эти и подобные им варианты начертания задаются следующими тегами:

- <B>текст жирный </B>
- <I>текст наклонный </I>
- <U> текст подчеркнутый </U>
- <BIG> большой текст</BIG>
- <SMALL> маленький текст </SMALL>
- <STRONG> увеличенный текст </STRONG>
- <STRIKE> перечеркнутый текст
- <SUB> подстрочный текст</SUB> ,
- <SUP> надстрочный текст</SUP>.

Пример создания веб-страницы с текстом разного начертания:

- <HTML>
- <HEAD>
- <TITLE> Стили шрифта </TITLE>
- </HEAD> <BODY>
- <BR> Текст <B> жирный </B>
- <BR> Текст <I> наклонный </I>
- <BR> Текст <U> подчеркнутый </U>
- <BR> Текст <TT> с моноширным шрифтом </TT>
- <BR> Текст <BIG> большой </BIG>
- <BR> Текст <SMALL> маленький </SMALL>
- <BR> Текст <STRONG> увеличенный </STRONG>

<BR> Текст <STRIKE> перечеркнутый </STRIKE>  
<BR> Текст <SUB> подстрочный </SUB>  
<BR> Текст <SUP> надстрочный <SUP>  
</BODY> </HTML>

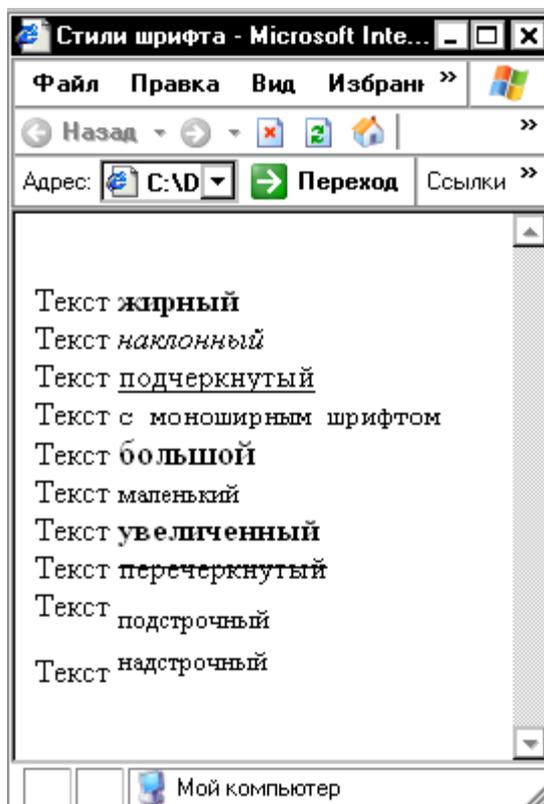


Рис. 5. Веб-страница с текстом разного начертания.

### Виды списков

Существует три основных вида списков в HTML-документе: нумерованный список; маркированный список; список определений.

Можно создавать вложенные списки, используя различные теги списков или повторяя одни внутри других. Для этого просто необходимо разместить одну пару тегов (открывающий и закрывающий) внутри другой.

### Нумерованные списки

В нумерованный список браузер автоматически вставляет номера элементов по порядку. Это означает, что если удалить один или несколько элементов нумерованного списка, остальные номера автоматически будут пересчитаны.

Нумерованные списки создаются с помощью парного тега <OL>. Каждый элемент списка начинается с тега <LI>.

Пример веб-страницы с нумерованным списком:

<HTML>

<HEAD>

<TITLE> Нумерованный список </TITLE>

</HEAD>

<H3> Правила создания веб страниц </H3>

<OL>

<LI> Веб-страницы не должны быть слишком широкими, чтобы пользователям не пришлось пользоваться прокруткой.

<LI>Каждая веб-страница должна иметь заголовок.

<LI>Не перегружайте страницу графикой.

<LI>Попытка визуально выделить все означает не выделить ничего.

<LI>Не забывайте житейскую мудрость: "Все гениальное - просто".

</OL>

</BODY>

</HTML>

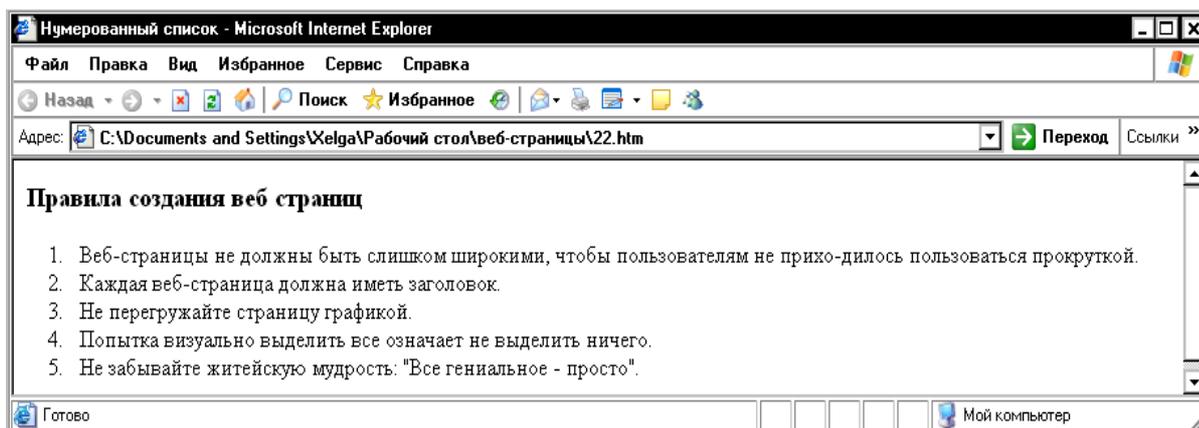


Рис. 11. Веб-страница с нумерованным списком

### Маркированные списки

Маркированные списки создаются с помощью парного тега <UL>. Каждый элемент списка начинается с тега <LI>. Тег <UL> может иметь атрибут TYPE:

<UL TYPE=disc|circle|square

Атрибут TYPE определяет внешний вид маркера. Доступные значения: disc — закрашенные круглые маркеры (вариант, заданный по умолчанию); circle — незакрашенные круглые маркеры; square — квадратные маркеры.

Пример веб-страницы с маркированным списком с маркером, заданным по умолчанию:

<HTML>

<HEAD>

<TITLE> Маркированный список </TITLE>

</HEAD>

<H3> Правила создания веб страниц </H3>

<UL>

<LI> Веб-страницы не должны быть слишком широкими, чтобы пользователям не приходилось пользоваться прокруткой.

<LI>Каждая веб-страница должна иметь заголовок.

<LI>Не перегружайте страницу графикой.

<LI>Попытка визуально выделить все означает не выделить ничего.

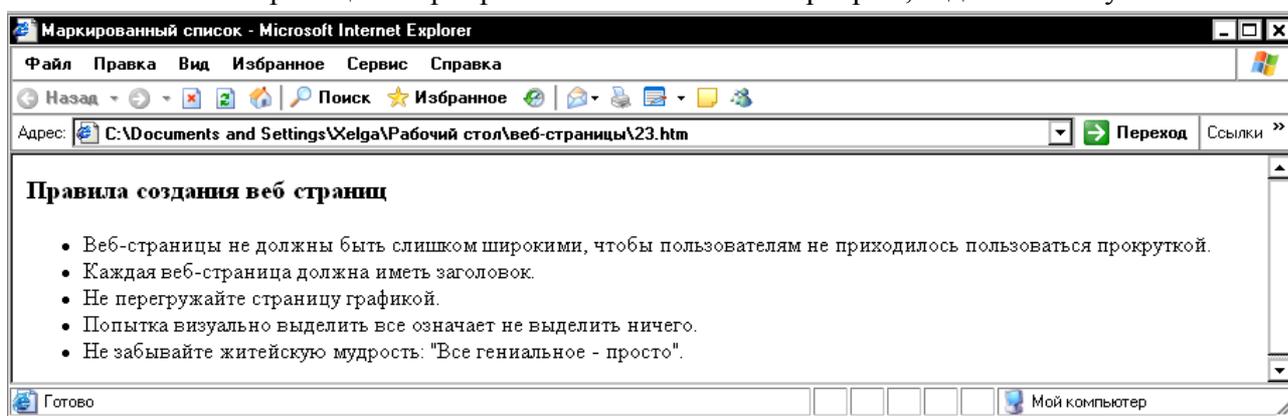
<LI>Не забывайте житейскую мудрость: "Все гениальное - просто".

</UL>

</BODY>

</HTML>

Рис. 12. Веб-страница с маркированным списком с маркером, заданным по умолчанию.



### Задания для выполнения

1. В Блокноте откройте файл start.htm
2. Отформатируйте текст, как показано ниже:

### Программное обеспечение ПК

**Программное обеспечение** - совокупность программ обработки данных.

*Программа* - упорядоченная последовательность команд (инструкций) компьютера для решения задачи.

Программный продукт - комплекс взаимосвязанных программ для решения определенной задачи массового спроса, подготовленный к реализации как любой вид промышленной продукции.

Интегрированное программное обеспечение — набор нескольких программных продуктов, функционально дополняющих друг друга, поддерживающих единые информационные технологии, реализованные на общей вычислительной и операционной платформе.

7. Добавьте два списка маркированный и нумерованный:

Области использования программных продуктов:

- аппаратная часть ПК;
- функциональные задачи различных предметных областей;
- технология разработки программ.

Классы программных продуктов:

1. системное программное обеспечение;
  2. пакеты прикладных программ;
  3. инструментарий технологии программирования.
8. Сохраните файл с именем format.htm;
4. Просмотрите страницу в обозревателе Internet Explorer.
5. Покажите работу преподавателю.

### **Тема 3 язык HTML: добавление графических элементов.**

#### **Размещение графики на веб-странице**

Графика позволяет значительно улучшить внешний вид и функциональность документов. Для вставки графики в веб-страницу используется тег <IMG>. Он имеет атрибуты SRC, ALT, LONGDESC, HEIGHT, WIDTH, USEMAP, ISMAP, ALIGN, BORDER, HSPACE, VSPACE.

Атрибут SRC. Атрибут SRC определяет месторасположение изображения, включаемого в состав документа. URL — обязательный параметр, который указывает браузеру, где находится рисунок. Рисунок должен храниться в графическом формате, например в формате GIF или JPG. Если графический файл находится в том же каталоге, что и содержащий его HTML-документ, достаточно указать только имя этого файла, например:

```
<IMG SRC=f11el.gif>
```

Если графический файл находится на том же сервере, что и содержащий его HTML-документ, но в другом каталоге, следует указать имя каталога и имя этого файла, например:

```
<IMG SRC=picture/f11el.gif>.
```

Если графический файл находится не на том сервере, на котором находится содержащий его HTML-документ, необходимо указать полный адрес этого файла, например:

```
<IMG SRC="http://www.uprlnt.ru/picture/f11el.gif">.
```

Атрибут HEIGHT. Необязательный атрибут HEIGHT определяет высоту рисунка в пикселах. Если данный параметр не указан, используется оригинальная высота рисунка. Этот параметр позволяет сжимать или растягивать изображения по вертикали, что дает возможность точнее контролировать внешний вид документа. Однако некоторые браузеры не поддерживают данный параметр. Иногда экранное разрешение на разных компьютерах может быть разным, поэтому при задании абсолютной величины графического объекта следует быть внимательным.

Атрибут WIDTH. Необязательный атрибут WIDTH используется для указания ширины рисунка и пикселах,

Атрибут ALIGN. Необязательный атрибут ALIGN используется для выравнивания объектов на экране. Доступные значения:

top — верхний край объекта выравнивается по верхнему краю строки;

middle — центр объекта выравнивается по базовой линии строки;

bottom — нижний край объекта выравнивается по базовой линии строки;

left — объект выравнивается по левому краю, при этом возможно обтекание объекта текстом;

right — объект выравнивается по правому краю, при этом возможно обтекание объекта текстом.

Если данный параметр не указан, большинство браузеров располагает изображение в левой части экрана, а текст — справа от него.

Атрибут BORDER. Атрибут BORDER задает толщину обрамления для изображения.

Атрибут VSPACE. Атрибут VSPACE позволяет задать размер в пикселах пустого пространства над и под изображением, чтобы текст «не наезжал» на изображение.

Атрибут HSPACE. Атрибут HSPACE позволяет задать размер в пикселах пустого пространства слева и справа от изображения, чтобы текст «не наезжал» на изображение.

### **Форматы графических файлов**

Большинство форматов файлов в компьютерной графике позволяет хранить информацию об изображении как о наборе точек. Точно так же (в виде набора точек) изображения выводятся на экран.

Обычно в Интернете используются два растровых формата — GIF и JPEG.

Пример создания веб-страницы с рисунком:

```
<HTML>
<HEAD>
<TITLE> Рисунок на веб-странице </TITLE>
</HEAD>
<BODY>
<IMG SRC="Осень.jpg" ALIGN=RIGHT>
<FONT SIZE=4>
<H2> Осень </H2>
<H3> А.С.Пушкин </H3>
<BR> Октябрь уж наступил-уж роща отряхает
<BR> Последние листы с нагих своих ветвей;
```

```
<BR> Дохнул осенний хлад-дорога промерзает,  
<BR> Журча еще бежит за мельницу ручей,  
<BR> Но пруд уже застыл; сосед мой поспешает  
<BR> В отъезжие поля с охотою своей,  
<BR> И страждут озими от бешеной забавы,  
<BR> И будит лай собак уснувшие дубравы.  
</BODY>  
</HTML>
```

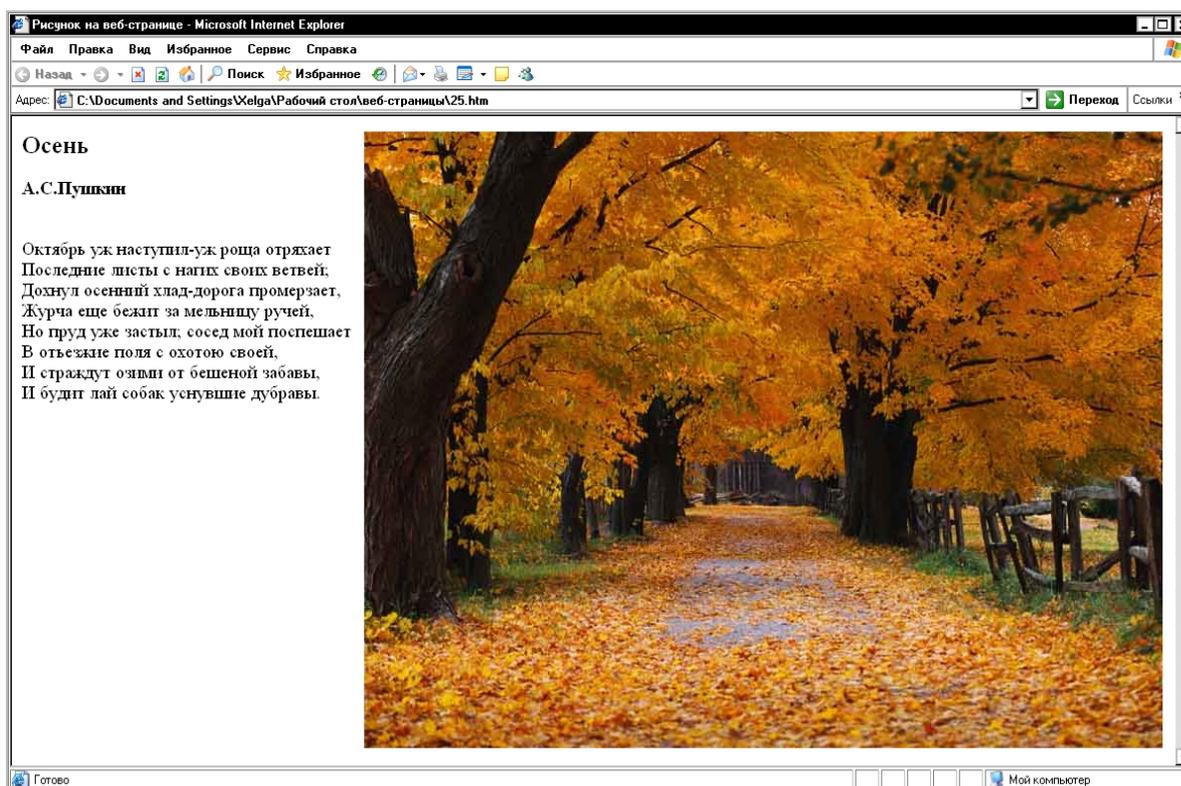


Рис. 14. Веб-страница с рисунком.

### Задание для выполнения

- 1.Создите веб-страницу.
- 2.На ней разместите:
  1. заголовок окна «Страница с картинкой1»;
  2. заголовок страницы «Системное программное обеспечение»;
  3. текст документа:

**Системное программное обеспечение** - совокупность программ и программных комплексов для обеспечения работы компьютера и компьютерных сетей.

**Базовое программное обеспечение** - минимальный набор программных средств, обеспечивающих работу компьютера.

**Сервисное программное обеспечение (программы-утилиты)** — программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя.

4. к тексту документа примените выравнивание;
5. выберите картинку, сохраните ее в своей папке;
6. на странице поместите рисунок (путь к рисунку можно не указывать, он находится в той же папке, что и ваша страница);
7. к рисунку примените выравнивание.
8. Сохраните файл с именем `gisunok1.htm` в своей папке.

3.Просмотрите страницу в обозревателе Internet Explorer.

4. Аналогично создайте еще две страницы.

5.На них разместите по картинке и текст (к тексту и к картинкам примените выравнивание) – на одной:

**Прикладное программное обеспечение.** Данный класс программных средств наиболее представлен, что обусловлено прежде всего широким применением средств компьютерной техники во всех сферах деятельности человека, созданием автоматизированных информационных систем различных предметных областей.

на другой:

**Инструментарий технологии программирования.** Это совокупность программ и программных комплексов, обеспечивающих технологию разработки, отладки и внедрения создаваемых программных продуктов. Инструментарий технологии программирования обеспечивает процесс разработки программ и включает специализированные программные продукты, которые являются инструментальными средствами разработчика.

6. Сохраните файлы соответственно с именами `gisunok2.htm`, `gisunok3.htm` в своей папке.
7. Просмотрите страницу в обозревателе Internet Explorer.
8. Покажите работу преподавателю.

#### **Тема 4: таблицы.**

##### **Создание таблиц**

Ячейки таблиц в языке HTML могут содержать любые HTML-элементы, в том числе заголовки, списки, текстовые абзацы, графику, а также элементы форм.

Основным тегом, описывающим таблицу, является парный тег `<TABLE>`. Все элементы таблицы должны находиться внутри тегов `<TABLE>` и `</TABLE>`. По умолчанию таблица не имеет обрамления и разделителей. Обрамление добавляется атрибутом `BORDER`. Размер рамки может быть фиксированным или автоматически согласовываться с размерами окна просмотра браузера и размерами текста и рисунков в ячейках. Помимо своего естественного назначения

(упорядочения текстовой информации), таблицы позволяют решать чисто дизайнерские задачи: выравнивать части фрагментов страниц друг относительно друга, размещать рядом рисунки и текст, управлять цветовым оформлением, разбивать текст на столбцы и т. д.

Тег <TABLE> имеет атрибуты ALIGN, BORDER, BACKGROUND, BGCOLOR, BORDERCOLOR, CELLPADDING, CELLSPACING, HSPACE, SPACE, COLSPEC, WIDTH.

Атрибут ALIGN. Если атрибут ALIGN располагается внутри тегов <CAPTION> и </CAPTION>, он определяет положение подписи таблицы и может принимать значения TOP (сверху, вариант заданный по умолчанию) и BOTTOM (снизу). Если атрибут ALIGN встречается внутри тегов <TR>...</TR>, <TH>...</TH> или <TD>...</TD>, он управляет выравниванием данных в ячейках по горизонтали и может принимать значения LEFT (слева), RIGHT (справа) или CENTER (по центру).

Атрибут BORDER. Атрибут BORDER определяет вид границ таблицы. Если данный атрибут присутствует, граница таблицы прорисовывается для всех ячеек и для таблицы в целом. Атрибут BORDER может принимать числовые значения, определяющие толщину границы, например:

BORDER=3

Атрибут BACKGROUND. Атрибут BACKGROUND позволяет создать фоновое изображение для всей таблицы:

BACKGROUND = URL

Атрибут BGCOLOR. Атрибут BGCOLOR задает фоновый цвет.

Атрибут BORDERCOLOR. Атрибут BORDERCOLOR задает цвет рамки. Используется только с атрибутом BORDER, например:

<TABLE BORDER="число" BORDERCOLOR="цвет">

Атрибут CELLPADDING. Атрибут CELLPADDING определяет расстояние от границ ячейки до ее содержимого в пикселях.

Атрибут CELLSPACING. Атрибут CELLSPACING определяет расстояние между ячейками в пикселях.

Атрибут HSPACE. Атрибут HSPACE задает свободное пространство слева и справа от таблицы в пикселях.

Атрибут VSPACE. Атрибут VSPACE задает свободное пространство сверху и снизу от таблицы в пикселях.

Атрибут COLSPEC. Атрибут COLSPEC позволяет создать столбцы фиксированной ширины. Значение ширины задается либо в символах, либо в процентах, например:

COLSPEC="20%"

Атрибут WIDTH. Атрибут WIDTH определяет ширину таблицы. Значение ширины задается либо в пикселях, либо в процентах от текущей ширины окна браузера, например:

WIDTH=256

WIDTH=100%

Если вводимый текст не помещается в ячейку, то в нее с помощью символов &nbsp; вводят неразрывный пробел. Это необходимо для прорисовки сетки таблицы.

### **Создание строк и столбцов таблицы**

Для создания строк и столбцов таблицы служат парные теги <TR> и <COL>.

Тег <TR> задает строку таблицы. Количество строк определяется количеством встречающихся пар тегов <TR> и </TR>. Строки могут иметь атрибуты ALIGN (выравнивание по горизонтали) и VALIGN (выравнивание по вертикали).

Тег <COL> используется для задания столбцов таблицы и может иметь атрибуты WIDTH, BGCOLOR, ALIGN, VALIGN и ID.

Атрибут WIDTH. Атрибут WIDTH задает ширину ячеек в столбце:

<COL WIDTH=число>

Атрибут BGCOLOR. Атрибут BGCOLOR задает цвет фона ячеек:

<COL BGCOLOR=цвет>

Атрибут ALIGN. Атрибут ALIGN задает режим горизонтального выравнивания содержимого внутри ячейки. Он может принимать значения LEFT (влево), CENTER (по центру) и RIGHT (вправо), например:

<COL ALIGN =значение>

Атрибут VALIGN. Атрибут VALIGN задает режим вертикального выравнивания содержимого внутри ячейки. Он может принимать значения MIDDLE (по середине) и TOP (вверх), например:

<COL VALIGN=значение>

Атрибут ID. Атрибут ID задает имя для ссылки, например:

<COL ID =имя>

### **Создание ячеек таблицы**

Для создания ячейки таблицы служит парный тег <TD> . Ячейка таблицы может быть описана только внутри строки таблицы. Каждая ячейка должна быть пронумерована номером столбца, к которому она относится. Если в строке для некоторых столбцов отсутствует одна или несколько ячеек, браузер отображает пустую ячейку. Расположение данных в ячейке по умолчанию определяется атрибутами ALIGN=LEFT и VALIGN=MIDDLE. Данное расположение может быть изменено как на уровне описания строки, так и на уровне описания ячейки.

### **Создание заголовка и подписи таблицы**

Для создания заголовка таблицы используется парный тег <TH>. Ячейка заголовка таблицы имеет ширину всей таблицы. Текст в данной ячейке имеет атрибуты BOLD и ALIGN=CENTER.

Для создания подписи таблицы используется парный тег <CAPTION>. Тег <CAPTION> должен присутствовать внутри тегов <TABLE> и </TABLE>, но вне описания строки или ячейки. Атрибут ALIGN определяет, где будет находиться подпись. По умолчанию атрибут ALIGN имеет значение TOP (сверху), но может быть установлен равным BOTTOM (снизу). Подпись всегда центрирована в рамках таблицы по горизонтали.

Пример создания таблицы без рамок из восьми строк и пяти столбцов с шириной равной ширине экрана, надписи в которой центрированы:

```
<HTML>
<HEAD>
<TITLE> Таблица без рамок, надписи в которой центрированы </TITLE>
</HEAD>
<BODY>
<H3> Простая таблица </H3>
<TABLE WIDTH=100%>
  <TR ALIGN=CENTER> <TD> Размер шрифта </TD> <TD> Типичная величина в пикселях
</TD></TR>
  <TR ALIGN=CENTER> <TD> 1 </TD> <TD> 8 </TD></TR>
  <TR ALIGN=CENTER> <TD> 2 </TD> <TD> 10 </TD></TR>
  <TR ALIGN=CENTER> <TD> 3 </TD> <TD> 12 </TD></TR>
  <TR ALIGN=CENTER> <TD> 4 </TD> <TD> 14 </TD></TR>
  <TR ALIGN=CENTER> <TD> 5 </TD> <TD> 18 </TD></TR>
  <TR ALIGN=CENTER> <TD> 6 </TD> <TD> 24 </TD></TR>
  <TR ALIGN=CENTER> <TD> 7 </TD> <TD> 36 </TD></TR>
</TABLE>
</BODY>
</HTML>
```

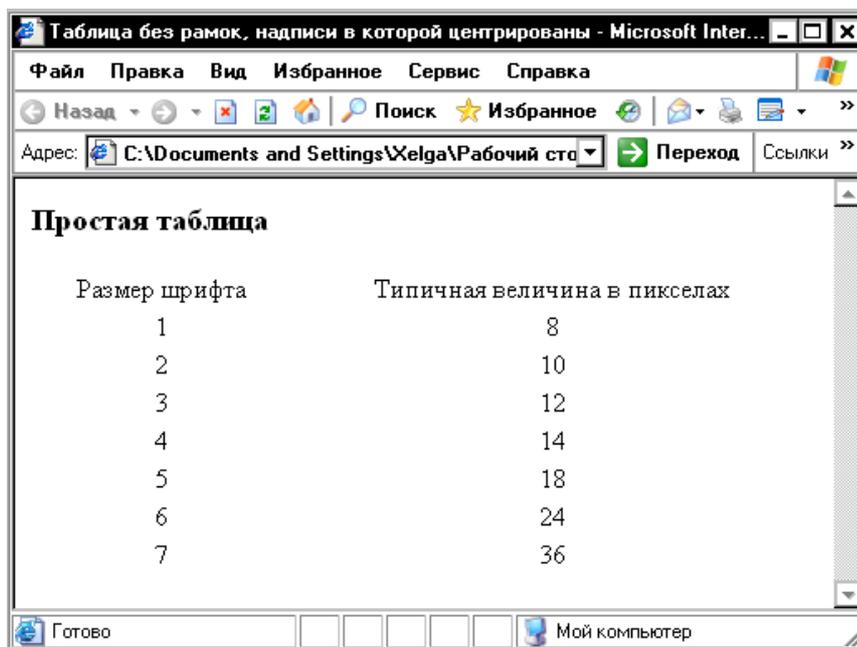


Рис. 15. Веб-страница с таблицей без рамок из восьми строк и пяти столбцов с шириной равной ширине экрана, надписи в которой центрированы.

## Теги группирования элементов таблиц

Для группирования элементов таблиц служат теги <COLGROUP>, <TBODY> и <TFOOT>.

### Группирование столбцов таблицы

Тег <COLGROUP> используется с целью группирования столбцов таблицы и может иметь атрибуты BGCOLOR, ALIGN, VALIGN, SPAN, WIDTH, ID.

Атрибут BGCOLOR. Атрибут BGCOLOR задает цвет фона ячеек, например:

```
<COLGROUP BGCOLOR=цвет>
```

Атрибут ALIGN. Атрибут ALIGN задает режим горизонтального выравнивания содержимого внутри ячейки. Он может принимать значения LEFT (влево), CENTER (по центру) и RIGHT (вправо), например:

```
<COLGROUP ALIGN=значение>
```

Атрибут VALIGN. Атрибут VALIGN задает режим вертикального выравнивания содержимого внутри ячейки. Он может принимать значения MIDDLE (по середине) и TOP (вверх), например:

```
<COLGROUP VALIGN =значение>
```

Атрибут SPAN. Атрибут SPAN задает количество столбцов в группе, например:

```
<COLGROUP SPAN=число>
```

Атрибут WIDTH. Атрибут WIDTH задает ширину ячеек в столбцах, например:

```
<COLGROUP WIDTH =число>
```

Атрибут ID. Атрибут ID задает имя для ссылки, например:

```
<COLGROUP ID=имя>
```

### Группирование строк таблицы

Для создания логически связанных групп строк в теле таблицы используется тег <TBODY>, например:

```
<TBODY><TR><TD>...<TD>...</TBODY>
```

У тега <TBODY> есть атрибуты BGCOLOR, ALIGN, VALIGN, которые имеют тот же смысл, что и одноименные атрибуты тега <COLGROUP>.

Для создания верхней шапки таблицы используется тег <THEAD>, например:

```
<THEAD><TR><TD>...<TD>...</THEAD>
```

У тега <THEAD> есть атрибуты BGCOLOR, ALIGN, VALIGN, ID, которые имеют тот же смысл, что и одноименные атрибуты тега <COLGROUP>.

Для создания нижней шапки таблицы используется тег <TFOOT>, например:

```
<TFOOT><TR><TD>...<TD>...</THEAD>
```

У тега <TFOOT> есть атрибуты BGCOLOR, ALIGN, VALIGN, ID, которые имеют тот же смысл, что и одноименные атрибуты тега <COLGROUP>.

Пример создания таблицы, в которой столбцы и строки сгруппированы:

```

<HTML>
<HEAD>
<TITLE> Таблица, в которой строки и столбцы сгруппированы </TITLE>
</HEAD>
<BODY>
<H3> Таблица </H3>
<TABLE BORDER=4>
<TR> <TD ROWSPAN=2> </TD> <TH COLSPAN=2> Среднее значение </TH></TR>
<TR> <TH> Рост </TH> <TH> Вес </TH> </TR>
<TR> <TD> Мужчины </TD> <TD ALIGN=CENTER> 184 </TD> <TD ALIGN=CENTER>
88 </TD> </TR>
<TR> <TD> Женщины </TD> <TD ALIGN=CENTER> 168 </TD> <TD ALIGN=CENTER>
63 </TD> </TR>
</TABLE>
</BODY>
</HTML>

```

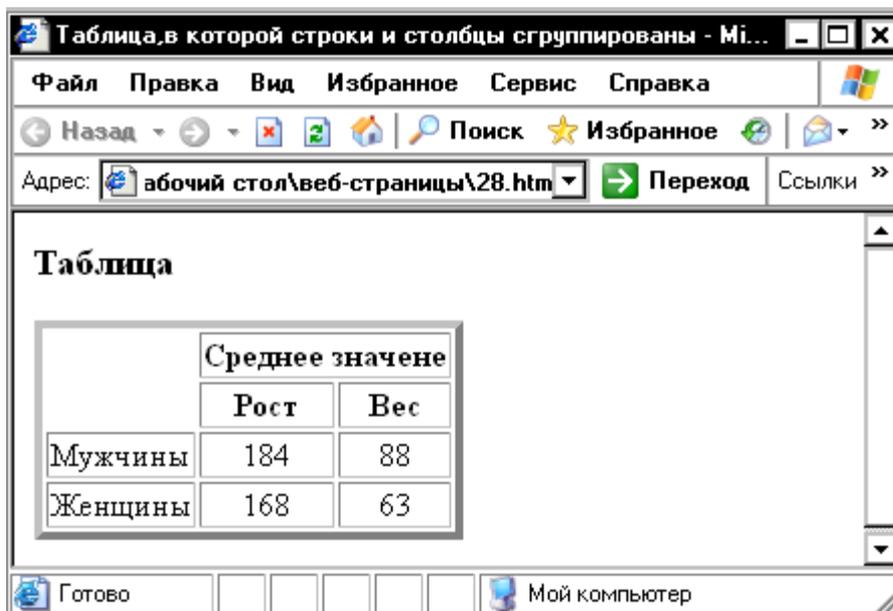


Рис. 16. Веб-страница с таблицей, в которой столбцы и строки сгруппированы.

### Основные атрибуты элементов таблицы

Ниже перечислены основные атрибуты элементов таблиц.

Атрибут ALIGN. Атрибут ALIGN управляет выравниванием содержимого ячеек. Допустимые значения LEFT (влево), RIGHT (вправо) и CENTER (по центру).

Атрибут BACKGROUND. Атрибут BACKGROUND задает фоновое изображение для ячеек строки:

BACKGROUND=URL

Атрибут BGCOLOR. Атрибут BGCOLOR задает фоновый цвет для ячеек строки.

Атрибут BORDER. Атрибут BORDER задает толщину рамки, обрамляющей ячейки строки. Нулевое значение данного атрибута означает отсутствие обрамления.

Атрибут VALIGN. Атрибут VALIGN встречается внутри тегов <TR>, <TH> и <TD>. Он определяет вертикальное выравнивание данных в ячейках. Доступные значения TOP (вверху), BOTTOM (внизу), CENTER (по центру), MIDDLE (по середине) и BASELINE (по базовой линии).

Атрибут NOWRAP. Атрибут NOWRAP говорит о том, что данные в ячейке не могут логически разбиваться на строки и должны быть представлены одной строкой.

Атрибут COLSPAN. Атрибут COLSPAN указывает, какое количество ячеек будет объединено по горизонтали с указанной ячейкой. По умолчанию значение этого атрибута равно 1.

Атрибут ROWSPAN. Атрибут ROWSPAN указывает, какое количество ячеек будет объединено по вертикали с указанной ячейкой. По умолчанию значение этого атрибута равно 1.

Атрибут RULES. Атрибут RULES определяет правила вывода линий между ячейками, например:

<TABLE RULES=ALL>

Доступные значения: Q NONE — нет линий (это значение используется по умолчанию);

GROUPS — линии отображаются только между группами строк (которые задаются тегами <THEAD>, <TFOOT> и <TBODY>) или группами столбцов (которые задаются тегами <COLGROUP и <COL>);

ROWS — линии отображаются только между строками;

COLS — линии отображаются только между столбцами;

ALL — линии отображаются между строками и столбцами.

Атрибут FRAME. Атрибут FRAME указывает, какие стороны фрейма, окружающего таблицу, должны быть видимыми, например:

<TABLE FRAME=VOID>

Доступные значения:

VOID — стороны невидимы (это значение используется по умолчанию);

ABOVE — видима только верхняя сторона;

BELOW — видима только нижняя сторона;

HSIDES — видимы только верхняя и нижняя стороны;

VSIDES — видимы только левая и правая стороны;

LHS — видима только левая сторона;

RHS — видима только правая сторона;

BOX — видимы все четыре стороны;

BORDER — как и в предыдущем случае, видимы все четыре стороны (визуальных различий между этими двумя значениями нет).

Атрибут ID. Атрибут ID задает имя для ссылки, например:

<TABLE ID =имя>

### Задания для выполнения

- 1.Создите веб-страницу.
- 2.На ней разместите
  1. заголовок окна «Страница с таблицей»;
  2. таблицу:

#### Классификация прикладного программного обеспечения

Класс прикладных программ	Подкласс прикладных программ
Проблемно-ориентированные ППП	Классификация по видам информационных систем и предметным областям
ППП автоматизированного проектирования	
ППП общего назначения	СУБД Сервер баз данных Генераторы отчетов Текстовые процессоры Табличные процессоры Средства презентационной графики Интегрированные пакеты
Методо-ориентированные ППП	Математическое программирование Статистическая обработка данных Сетевые методы и модели для решения управленческих задач
Офисные ППП	Коммуникационные системы Органайзеры Переводчики, средства проверки орфографии и распознавания текста
Программные средства мультимедиа	
Настольные издательские системы	
Системы искусственного интеллекта	

6. Сохраните файл с именем tab.htm в своей папке.
7. Просмотрите страницу в обозревателе Internet Explorer.
8. Покажите работу преподавателю.

### Тема 5: Гипертекстовые ссылки

#### Гипертекстовые ссылки

Гипертекст — информационная структура, обеспечивающая навигацию посредством гипертекстовых ссылок. Гипертекстовыми ссылками называются фрагменты текста или изображения, при активизации которых отображаются связанные с ними документы.

Гипертекстовая ссылка задается тегами <A> и </A>. Она имеет несколько атрибутов, наличие одного из двух (HREF или NAME) обязательно. Первый указывает на ссылку за пределы данного документа, например:

<A HREF="http://www.имя\_сайта.зона/.../имя\_файла.htm"> название ссылки </A> Второй позволяет перейти к определенным образом отмеченному (с помощью так называемого якоря, или закладки) месту того же самого документа, например:

<A NAME="якоря"> название ссылки </A>

Ссылки можно использовать для перемещения как по одному документу так и от одного документа к другому или для запроса файла по протоколу FTP (File Transfer Protocol - протокол передачи файлов) для отображения его браузером. В ссылке включается URL (Uniform Resource Locator- унифицированный указатель ресурса) – адрес идентифицирующий другие локальные или удаленные документы. Однако HTML не поддерживает возврат на предыдущую ссылку, если перемещение происходит внутри документа.

Пример документа, содержащего ссылки на другие веб-страницы текущего веб-узла:

```
<HTML>
<HEAD>
<TITLE> Интернет-технологии </TITLE>
</HEAD>
<BODY>
<BODY BACKGROUND=FFFFFF>
<CENTER> <H1> Лабораторные работы по курсу "Интернет-технологии" </H1>
</CENTER>
<HR NOSHADE SIZE=10 WIDTH="100%" ALIGN=CENTER>
<P> <H2> <A HREF="лаб раб №1.htm"> Лабораторная работа №1 </A> </P>
<P> <H2> <A HREF="лаб раб №2.htm"> Лабораторная работа №2 </A> </P>
<P> <H2> <A HREF="лаб раб №3.htm"> Лабораторная работа №3 </A> </P>
</BODY></HTML>
```

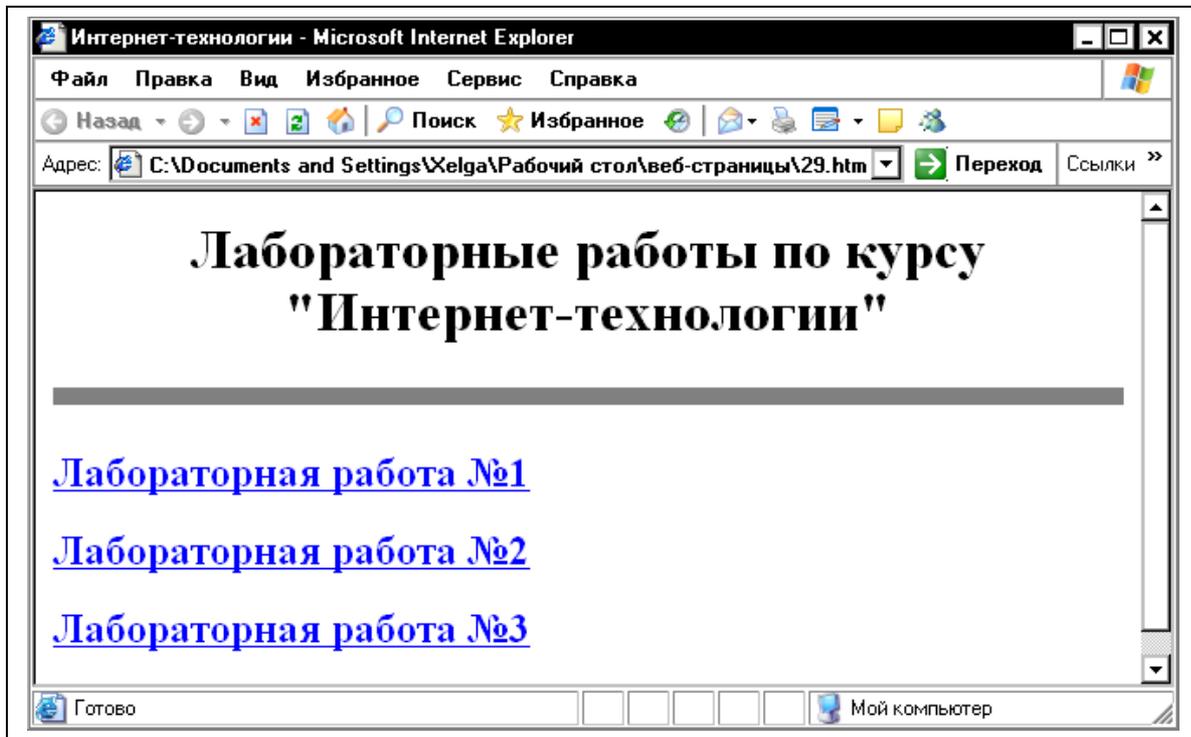


Рис. 17. Веб-страница, содержащая ссылки на другие веб-страницы текущего веб-узла.

Пример веб-страницы с якорями:

```

<HTML>
<HEAD>
<TITLE> Пример якоря </TITLE>
</HEAD>
<BODY>
<BODY BACKGROUND=FFFFFF>
<CENTER>
<H2> Оглавление </H2> </CENTER>
<UL> <LI> <A HREF="#glava1"> Глава 1 </A> </LI>
<LI> <A HREF="#glava2"> Глава 2 </A> </LI> </UL>
<P> <A NAME="glava1"> </A> Глава 1 </P>
<IL> <P> Текст главы 1 </P> </UL>
<P> <A NAME="glava2"> </A> Глава 2 </P></UL>
<IL> <P> Текст главы 1 <BR> </P>
</BODY>
</HTML>

```

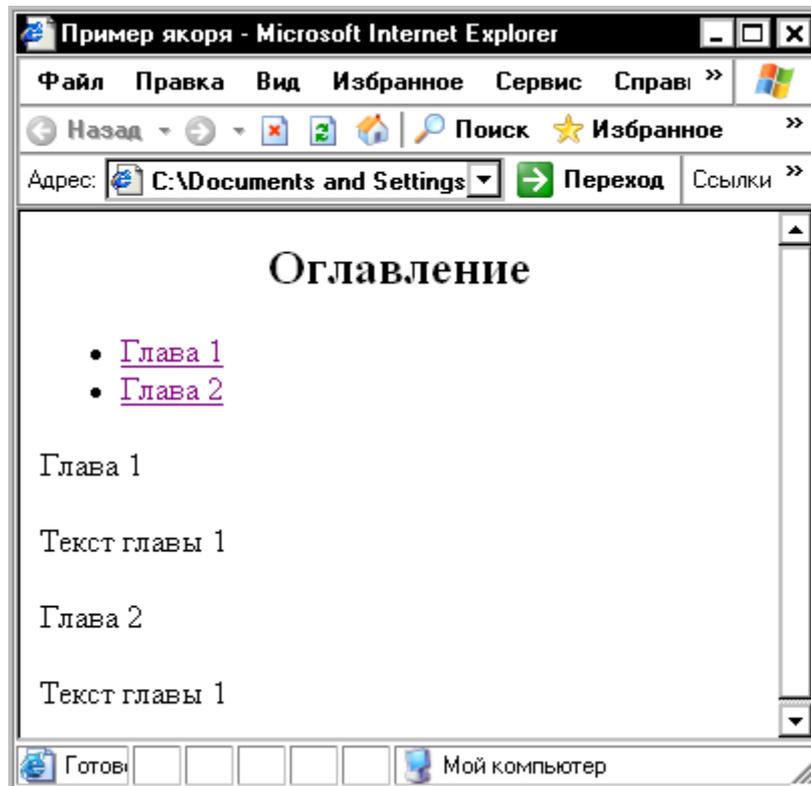


Рис. 18. Веб-страница с якорями.

### Задания для выполнения

1. В Блокноте откройте файл format.htm
2. Разместите гипертекстовые ссылки на файлы gisunok2.htm, gisunok3.htm gisunok1.htm (путь к файлам можно не указывать: они все в одной папке).
3. Сохраните файл в своей папке.
4. На веб-странице «Страница с картинкой1» (файл gisunok1.htm) разместите якоря. Сохраните изменения.
4. Просмотрите страницу в обозревателе Internet Explorer.
5. Покажите работу преподавателю.

### Тема 6: представление документов на панелях окна.

#### Фреймы

Фреймы позволяют разбивать веб-страницы на подокна, причем каждое подокно (фрейм) может иметь собственные полосы прокрутки и функционировать независимо от других подокон либо, наоборот, — управлять их содержанием. Фреймы могут значительно улучшить внешний вид веб-страниц и повысить их функциональность.

#### Свойства фреймов

Каждый фрейм имеет свой URL-адрес, что позволяет обращаться к нему и загружать его независимо от других фреймов.

Каждый фрейм имеет собственное имя (задаваемое атрибутом NAME), позволяющее обратиться к нему из другого фрейма.

Размер фрейма может быть изменен пользователем прямо на экране мышью (если это не запрещено с помощью специального параметра).

В отдельном фрейме, который будет все время оставаться на экране, можно разместить статическую информацию, которую автор считает необходимым постоянно показывать пользователю. Это может быть графический логотип фирмы, соглашение об авторских правах, набор управляющих кнопок, оглавление всех или части веб-документов, содержащихся на веб-сервере, что позволит пользователю быстро находить интересующую его информацию.

### Наборы фреймов

Формат документа, использующего фреймы, внешне очень напоминает формат обычного документа, только вместо тега <BODY> используется парный тег <FRAMESET>:

```
<HTML>
<HEAD>...</HEAD>
<FRAMESET>...</FRAMESET>
</HTML>
```

То есть перед тем как размещать на веб-страницы фреймы, с помощью тега <FRAMESET> создают так называемый набор фреймов, содержащий информацию о числе фреймов, их относительных и абсолютных размерах.

Общий синтаксис набора фреймов:

```
<FRAMESET COLS="value" | ROWS="value">
  <FRAME SRC="url">
  <FRAME ...>...
</FRAMESET>
```

Общий контейнер <FRAMESET> описывает все фреймы, на которые делится экран. Можно разделить экран на несколько вертикальных или несколько горизонтальных фреймов. Для описания каждого фрейма в отдельности служит тег <FRAME>.

Между тегами <FRAMESET> и </FRAMESET> могут находиться теги <FRAME>, вложенные теги <FRAMESET> и </FRAMESET>, а также контейнер из тегов <NOFRAME> и </NOFRAME>, который позволяет продублировать документ для браузеров, не поддерживающих фреймы.

Тег <FRAMESET> допускает вложения и может иметь атрибуты COLS, ROWS, BORDER, FRAMEBORDER, причем атрибуты ROWS и COLS взаимоисключающие.

Атрибут COLS. Атрибут COLS задает количество и размеры столбцов в создаваемом наборе фреймов. Значения ширины различных фреймов перечисляются в кавычках через запятую в пикселах, в процентах (%) или в связанных масштабных коэффициентах (\*), например:

```
<FRAMESET COLS ="число, число%, число*">
```

Атрибут ROWS. Атрибут ROWS задает количество и размеры строк в создаваемом наборе фреймов в пикселах, в процентах (%) или в связанных масштабных коэффициентах (\*), например:

```
<FRAMESET ROWS="число, число%; число*">
```

Количество фреймов определяется количеством значений в списке. Общая сумма значений высоты всех фреймов (при любом способе задания значений атрибута) должна составлять высоту всего окна. Отсутствие атрибута ROWS означает один фрейм высотой с окно браузера.

Атрибут BORDER. Атрибут BORDER задает толщину обрамления в пикселах для всех фреймов, входящих в данный набор фреймов, например:

```
<FRAMESET BORDER="число">
```

Атрибут FRAMEBORDER. Атрибут FRAMEBORDER позволяет отменить вывод обрамления у фреймов независимо от значения атрибута BORDER. Значение 1 соответствует наличию, а 0 — отсутствию обрамления, например:

```
<FRAMESET FRAMEBORDER="1">
```

```
<FRAMESET FRAMEBORDER="0">
```

Задание значений атрибута ROWS или COLS в пикселах — далеко не самый лучший способ описания высоты фрейма, поскольку пользователи могут произвольным образом менять размер рабочего поля браузера, не говоря уже о различных разрешениях экрана. Если вы все же выбрали данный способ описания размера, настоятельно рекомендуется сочетать его с каким-либо другим, чтобы в результате получилось 100-процентное заполнение окна браузера.

Если значения атрибута ROWS или COLS указываются в процентах (от 1% до 100%) и общая сумма размеров описываемых фреймов превышает 100%, то размеры всех фреймов пропорционально уменьшаются, чтобы в результате получилась сумма в 100 %. Если, соответственно, сумма меньше 100 %, то размеры пропорционально увеличиваются.

Задание одного из значений атрибута ROWS или COLS с единичным масштабным коэффициентом (в этом случае указывается просто звездочка без числа) означает, что все оставшееся место будет принадлежать данному фрейму. Если единичные масштабные коэффициенты указываются для двух или более фреймов, то оставшееся пространство делится поровну между этими фреймами. Если перед звездочкой стоит цифра, то она задает масштабный коэффициент для данного фрейма (во сколько раз он должен быть больше аналогичного фрейма, описанного звездочкой без числа).

Рассмотрим примеры.

```
COLS="3*.*.*"
```

Эта запись говорит о том, что будет создано три фрейма, причем первый фрейм займет 3/5 свободного пространства по горизонтали, а два других — по 1/5.

```
COLS="50%.50%"
```

Такая запись означает деление области просмотра на две равные части.

```
COLS="20%.80%"
```

Эта запись означает деление области просмотра на неравные части (слева 20, справа 80 %):

```
ROWS="100.20%.*"
```

Данная запись означает деление области просмотра на три части. Первой (верхней) отведено 100 пикселей, второй (средней) — 20 % оставшегося пространства, третьей (нижней) — все остальное

Фреймы поддерживаются не всеми браузерами. Чтобы пользователи таких браузеров могли видеть содержимое веб-страницы с фреймами, используется парный тег `<NOFRAMES>`. Этот тег помещается внутри контейнера `<FRAMESET>`, и все, что находится между тегами `<NOFRAMES>` и `</NOFRAMES>`, игнорируется браузерами, поддерживающими фреймы. То есть разработчику веб-страницы нужно написать код, дублирующий содержимое фреймов другими средствами, и поместить этот код между тегами `<NOFRAMES>` и `</NOFRAMES>`, тогда все пользователи смогут увидеть его веб-страницу.

Пример веб-страницы с набором фреймов:

```
<HTML> <HEAD> <TITLE> Фреймы </TITLE> </HEAD>
<FRAMESET COLS="60%,*">
<FRAMESET ROWS="60%,*">
<FRAMESET COLS="50%,*">
<FRAME SRC="текст №1.htm">
<FRAME SRC="текст №2.htm"> </FRAMESET>
<FRAME SRC="текст №3.htm"> </FRAMESET>
<FRAME SRC="текст №4.htm"> </FRAMESET> <HTML>
```

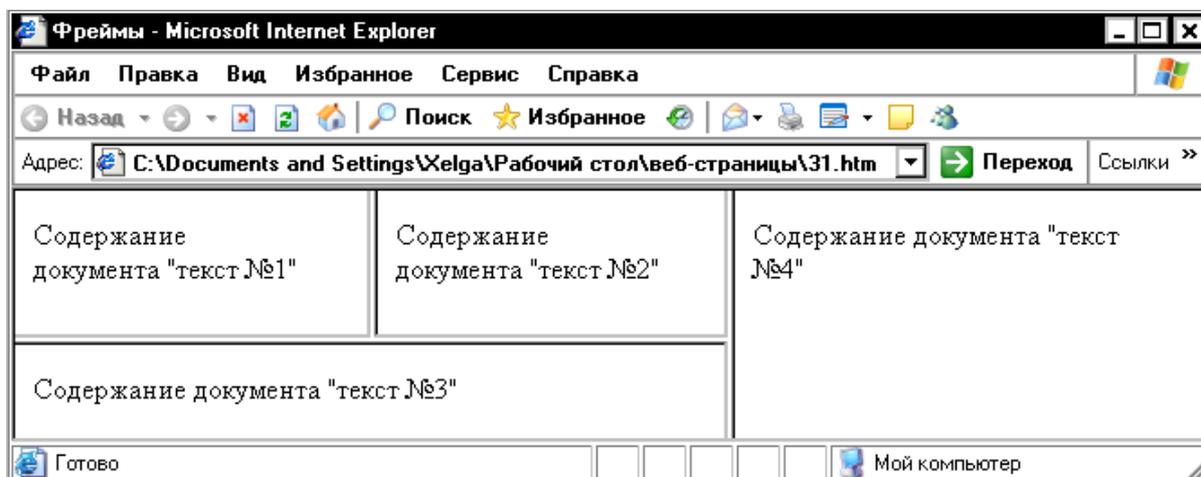


Рис. 19. Веб-страница с фреймами.

## Вставка фрейма в документ с набором фреймов

Как уже отмечалось, для вставки в документ отдельного фрейма служит непарный тег `<FRAME>` с атрибутами `NAME`, `SRC`, `MARGINWIDTH`, `MARGINHEIGHT`, `SCROLLING`, `NORESIZE`, `FRAMEBORDER`.

Атрибут `NAME`. Атрибут `NAME` задает имя фрейма. Имя позволяет обратиться к этому фрейму в программе (это делается с помощью атрибута `TARGET`), например загрузить в него содержимое нового HTML-файла. Синтаксис:

```
NAME="имя"
```

Атрибут `SRC`. Атрибут `SRC` задает документ, который должен отображаться внутри данного фрейма, например:

```
<FRAME SRC="10.htm">
```

Если атрибут `SRC` отсутствует, отображается пустой фрейм.

Атрибут `MARGINWIDTH`. Атрибут `MARGINWIDTH` позволяет указать величину вертикальных разделительных полос между фреймами:

```
MARGINWIDTH="value"
```

Значение `value` указывается в пикселах и не может быть меньше единицы. Значение, устанавливаемое по умолчанию, зависит от используемого браузера.

Атрибут `MARGINHEIGHT`. Атрибут `MARGINHEIGHT` позволяет указать величину горизонтальных разделительных полос между фреймами:

```
MARGINHEIGHT="value"
```

Значение `value` указывается в пикселах и не может быть меньше единицы.

Атрибут `SCROLLING`. Атрибут `SCROLLING` управляет выводом полос прокрутки фрейма:

```
SCROLLING="yes | no | auto"
```

Значение `yes` указывает, что фрейм будет иметь полосы прокрутки в любом случае, значение `no` — что полос прокрутки не будет. Значение `auto`, заданное по умолчанию, говорит о том, что полосы прокрутки появятся только при необходимости.

Атрибут `NORESIZE`. Атрибут `NORESIZE` позволяет создавать фреймы без возможности изменения их размеров. По умолчанию размер фрейма можно изменить мышью так же просто, как и размер окна Windows. Атрибут `NORESIZE` запрещает такую возможность. Если у одного из фреймов установлен атрибут `NORESIZE`, это лишает пользователя возможности перетаскивать границы не только данного фрейма, но и прилегающие к нему границы других фреймов.

Атрибут `FRAMEBORDER`. Атрибут `FRAMEBORDER` задает обрамление у фреймов и может принимать значения 1 (есть обрамление) или 0 (нет обрамления), например:

```
<FRAME FRAMEBORDER="1">
```

```
<FRAME FRAMEBORDER="0">
```

### Задания для выполнения

1. Создайте веб-страницу с фреймами.
2. С помощью атрибутов тега `FRAMESET` разделите экран на фреймы.

4. С помощью тега FRAME разместите содержание файлов start.htm, risunok1.htm, risunok2.htm, risunok3.htm в подокнах (путь к веб страницам можно не указывать, они находятся в одной папке).
5. Просмотрите документ в обозревателе Internet Explorer.
6. Покажите выполненную работу преподавателю.

## MathCAD

MathCAD – программное средство, среда для выполнения на компьютере разнообразных математических и технических расчетов, предоставляющая пользователю инструменты для работы с формулами, числами, графиками и текстами, снабженная простым в освоении графическим интерфейсом. (Графический интерфейс – совокупность способов взаимодействия пользователя с программой с помощью пиктограмм, диалоговых окон, меню и других инструментов, расположенных на экране.)

Пользовательский интерфейс MathCAD создан так, что пользователь, имеющий элементарные навыки работы с Windows-приложениями, может сразу начать работу с MathCAD. Интерфейс системы внешне очень напоминает интерфейс широко известных текстовых процессоров Word под Windows. С назначением элементов интерфейса можно ознакомиться, установив на соответствующий элемент (например кнопку) курсор мыши и задержав его на пару секунд, — тут же появится окошко, в котором дается краткая поясняющая надпись о назначении указанного элемента интерфейса.

Сразу после запуска система готова к созданию документа с необходимыми пользователю вычислениями.

## Меню Mathcad

На рисунке 1 представлено окно Mathcad, в терминологии Windows окно приложения:

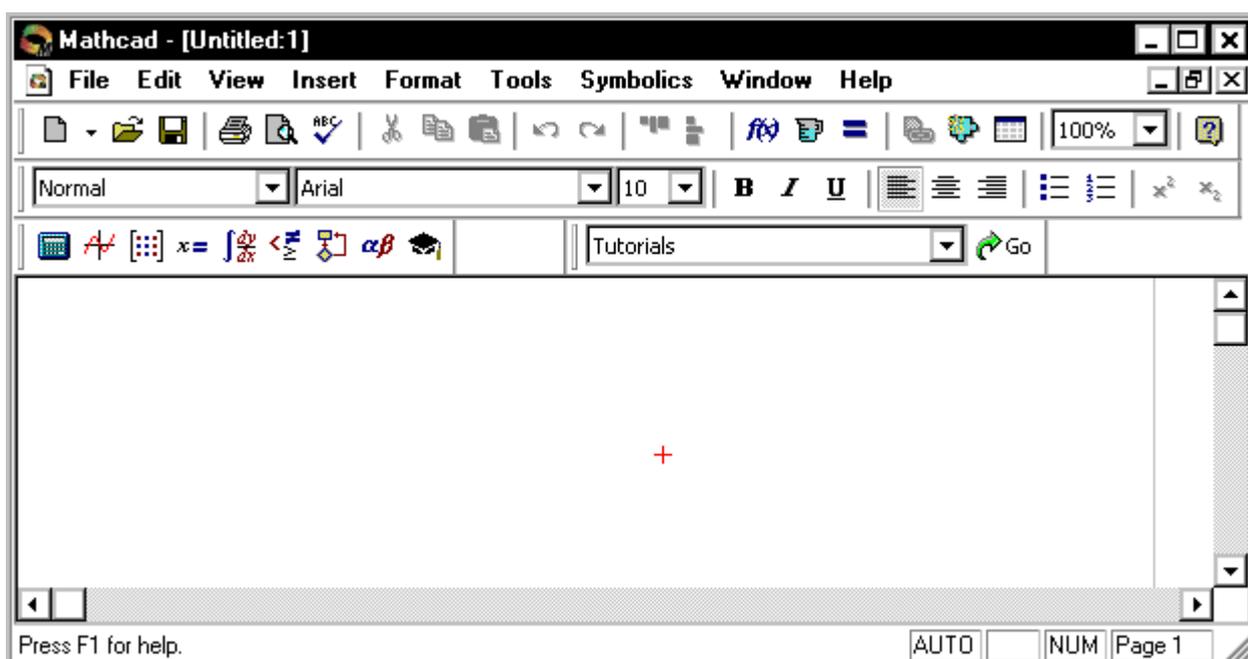


Рис. 1.

Это пространство, где размещены все требуемые для работы инструменты и рабочий документ, то есть пространство, в котором будут выводиться команды и выражения для вычислений и в котором будут отображены результаты вычислений.

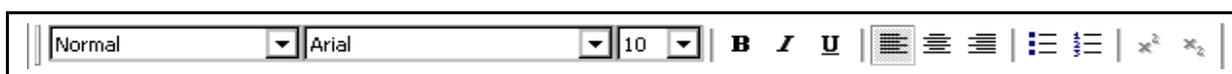
Верхняя строка окна – стандартная строка Windows-приложений. В ней слева приведено имя приложения Mathcad, затем имя файла, в котором сохраняются результаты работы, а справа три функциональные кнопки для работы с окнами Windows , соответственно означающие: свернуть, развернуть на полный экран и закрыть окно приложения.

Вторая строка экрана — строка меню:



Щелчок по пункту меню открывает ниспадающее меню со списком доступных в данном пункте меню операций.

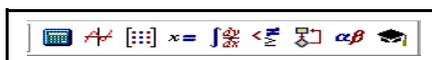
Следующие три строки окна содержат панели инструментов, часть из которых — стандартные для windows-приложений операции работы с файлами и текстом:



другая часть — специфические функции Mathcad:



В отдельной строке обычно располагают панель инструментов для выполнения математических операций, которую будем называть панелью математических инструментов или панелью математических операций:



Математические операции в Mathcad разделены на группы и каждая кнопка панели математических инструментов открывает доступ к определенной группе операций – щелчок по кнопке этой панели открывает другую, дополнительную, панель, на которой собственно и расположены кнопки математических операций соответствующей группы (подобно тому, как щелчок по пункту меню открывает ниспадающее меню).

Под строками панелей инструментов находится окно рабочего документа Mathcad — пространство, в котором располагаются все введенные команды и выражения, куда Mathcad выводит результаты вычислений и графики и где размещаются текстовые комментарии. Содержимое этого окна можно редактировать, форматировать, сохранять в файлах на диске,

печатать и др. Окно снабжено двумя полосами прокрутки — вертикальной и горизонтальной. С их помощью можно просматривать фрагменты окна, находящиеся вне пределов экрана.

И наконец, последняя, нижняя строка окна – строка состояния:



В ней приведены (слева направо) рекомендации к дальнейшим действиям, текущее состояние среды и номер отображенной на экране страницы рабочего документа.

В приведенном примере в строке состояния указано, что на экране отображена первая страница рабочего документа, система находится в режиме автоматических вычислений (AUTO) и программа рекомендует пользователю обратиться за помощью к справке, нажав на клавиатуре клавишу <F1>.

## МЕНЮ MATHCAD

Большинство вычислений в Mathcad можно выполнить тремя способами:

- выбором операции в меню;
- с помощью кнопочных панелей инструментов;
- обращением к соответствующим функциям.

Во всех трех случаях соблюдается один и тот же порядок действий:

выбор операции производится щелчком мыши по пункту меню или по кнопке в панели инструментов (иногда выбор производится в ниспадающем меню или в дополнительной панели); когда операция выбрана, в рабочем документе открывается окно диалога или поле ввода и пользователь вводит необходимую информацию в окне диалога или заполняет помеченные поля в поле ввода.

Кратко рассмотрим содержание некоторых пунктов меню.

**File** – работа с файлами, сетью Internet и электронной почтой;

**Edit** – редактирование документов;

**View** – изменение средств обзора и включения/выключения элементов интерфейса (Если одна из строк меню **Toolbar** (Панель инструментов), **Status Bar** (Строка состояния) или **Ruler** (Линейка) помечена символом  $\surd$ , то на экране размещается соответствующая панель. Пункт меню **Toolbar** открывает доступ к панелям инструментов: **Standard** (Стандартная), **Formatting** (Форматирование), **Math** (Математическая), а также доступ ко всем панелям математических инструментов: **Calculator** (Калькулятор), **Graph** (Графики), **Matrix** (Матрицы), **Evaluation** (Вычисления), **Calculus** (математический анализ), **Boolean** (Логические функции), **Programming** (Программирование), **Greek** (Греческие буквы), **Symbolic** (Символьные вычисления), **Modifier** (Преобразование типа);

**Insert** – установка вставок объектов и их шаблонов (включая графику);

**Format** – изменение формата (параметров) объектов;

**Symbolic** – выбор операций символьного процессора;

**Window** – управление окнами системы;

**Books** – работа с электронными книгами;

**Help** – работа со справочной базой данных о системе.

#### 1.4. Использование инструментальных и наборных панелей

Работа с документами MathCAD обычно не требует обязательного использования возможностей главного меню, т. к. основные из них дублируются кнопками быстрого управления. Их можно выводить на экран или убирать с него с помощью соответствующих опций позиции **View** главного меню Windows.

Чаще всего используются две такие панели: панель инструментов (дублирующая ряд наиболее распространенных команд и операций) и панель форматирования для выбора типа и размера шрифтов и способа выравнивания текстовых комментариев.

Полное описание меню можно найти во встроенном справочнике по работе с пакетом.

#### Панели инструментов MathCAD

Работа с документами MathCAD обычно не требует обязательного использования возможностей главного меню, т. к. основные из них дублируются кнопками быстрого управления. Их можно выводить на экран или убирать с него с помощью соответствующих опций позиции **View** главного меню Windows.

Чаще всего используются две такие панели: панель математических инструментов– Панель Math (дублирующая ряд наиболее распространенных команд и операций) и панель форматирования для выбора типа и размера шрифтов и способа выравнивания текстовых комментариев.

**Панель Math.** Здесь расположены кнопки вычислительных операций и операций конструирования вычислительных процедур. Панель математических инструментов содержит девять кнопок, каждая из которых открывает панель математических операций. Ниже перечислены кнопки панели математических операций.

На рисунке 2 показаны кнопки вызова палитр математических операций.

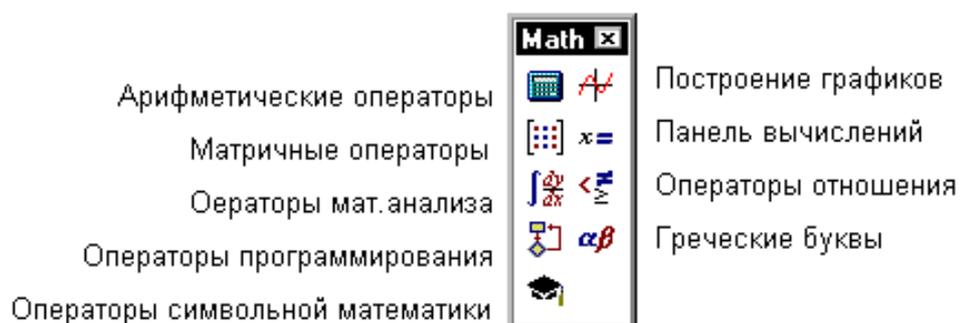


Рис. 2. Панель Math

## Основы работы с блоками документов

Документ состоит из отдельных блоков. Они могут быть различного типа: тексты (комментарии), формулы, графики, таблицы и т. д. Каждый блок занимает в текущем окне определенную область прямоугольной формы. Блоки исполняются строго последовательно — слева—направо и сверху—вниз.

### Выделение блоков документов

Для выделения областей блоков можно проделать следующий эксперимент. Поместите курсор мыши в нерабочую часть экрана и нажмите ее левую клавишу. Перемещая мышью по диагонали при удержанной нажатой клавише мыши, вы увидите, как на экране пунктирными черными линиями выделяется прямоугольник. Добейтесь, чтобы в него попало несколько объектов — формул, таблиц, графиков. Все они тут же окажутся, обведены прямоугольниками из пунктирных линий.

Эти прямоугольники являются обычно невидимыми границами областей объектов. Несколько выделенных таким образом объектов образуют блок выделенных объектов. Он ведет себя как единое целое — может перемещаться мышью, копироваться в буфер и стираться.

### Расположение блоков в документе

Расположение блоков в документе имеет принципиально важное значение. Как уже отмечалось, их выполнение происходит справа—налево и сверху—вниз. Поэтому блоки не должны взаимно перекрываться. Указанный порядок исполнения блоков означает, что, например, при построении графика функции или таблицы ее значений вначале должны исполняться блоки, задающие саму функцию и пределы изменения аргумента, а уже затем блок, задающий вывод таблицы или построение графика функции. Все это есть следствие работы системы в интерпретирующем режиме.

Доступная область окна редактирования по горизонтали превышает видимую на экране. При этом область окна делится длинной линией на две части, левую и правую. Горизонтальными линиями документ делится на страницы.

Можно разместить некоторые вспомогательные вычисления в правой, невидимой, части окна, а в левую, видимую, вынести задание исходных данных и вывод результатов расчетов. Этот прием часто используется в пакетах применения системы, а также в учебно-демонстрационных программах.

## Понятие о входном языке системы MathCAD

Общение пользователя с системой MathCAD происходит на некотором промежуточном математически ориентированном языке **визуального программирования** — входном языке. Многие математические записи в этом языке вводятся просто выводом шаблонов

соответствующих операторов и функций. Этот язык настолько приближен к обычному математическому языку описания вычислительных задач, что практически не требует их программирования. Нужно лишь точное описание алгоритма решения задачи.

**Операторы** — это специальные символы, указывающие на выполнение тех или иных операций над данными — **операндами**. Последние могут быть представлены константами или переменными — объектами с именами, хранящими данные определенного типа и значения. Наиболее известны арифметические операторы, например сложения +, вычитания -, умножения \*, деления /, вычисления квадратного корня и др. Операторы могут вводиться и с помощью наборных панелей.

**Функция** — объект входного языка, имеющий имя и параметры, указываемые в круглых скобках. Имя функции отождествляется с соответствующей математической функцией — например  $\sin(x)$  — это функция вычисления синуса аргумента  $x$ . Отличительной чертой функции является **возврат значения** (результата вычисления функции) в ответ на обращение к ней.

Операторы и функции используются для создания **математических выражений** — формул, которые могут вычисляться в численном или символьном виде.

С **переменными** связано понятие **присваивания** им значений. Символ присваивания := отличен от обычно используемого в математике знака равенства =. В системах MathCAD знак = используется как знак вывода результатов вычислений, а для присваивания переменным значений используется знак := (например  $a:=3$  присваивает переменной  $a$  значение числа 3). Наконец, для обозначения отношения величин  $x$  и  $y$  как равенства используется жирный знак равенства.

### Работа с формульным редактором

Фактически система MathCAD интегрирует в себе три редактора: формульный, текстовый и графический. Для запуска формульного редактора достаточно установить курсор мыши в любом свободном месте окна редактирования и щелкнуть левой клавишей. Появится визир в виде маленького красного крестика. Его можно перемещать клавишами перемещения курсора. Визир не нужно путать с курсором мыши — он живет своей жизнью и имеет вид жирной наклонной стрелки.

Визир указывает место, с которого можно начинать набор формул — вычислительных блоков. Щелчок левой клавиши мыши устанавливает визир на место, указанное острием стрелки курсора мыши.

### Простейшие арифметические операции

Вычислим значения арифметического выражения  $65 + \frac{13}{4}$ .

Щелкнем мышью по любому месту в рабочем документе — в поле появится крестик, отмечающий позицию, с которой начинается ввод.

Введем с клавиатуры символы в следующей последовательности:  $65 + \frac{13}{4}$

Введем с клавиатуры знак равенства, нажав клавишу =. MathCAD вычисляет значение выражения и выводит справа от знака равенства результат:

$$65 + \frac{13}{4} = 68.25$$

### Определение и вычисление значения функции в точке. Построение таблицы значений функции.

Определим функцию  $f(x) := \frac{x+1}{x^2+1}$ , вычислим ее значение при  $x=1.2$  и построим таблицу значений функции для  $x \in [0, 10]$  с шагом 1.

Щелкнем по свободному месту в рабочем документе, введем с клавиатуры  $f(x) = \frac{x+1}{x^2+1}$ .

Щелкнем по свободному месту в рабочем документе и введем с клавиатуры  $f(1.2) =$

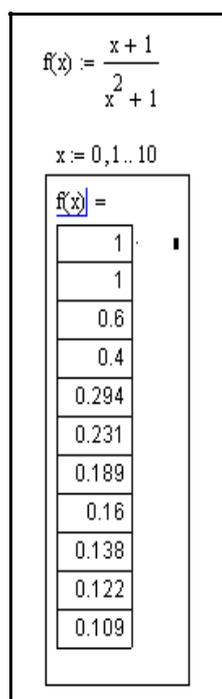
Сразу после ввода знака равенства в рабочий документ выводится значение функции  $f(x)$  при  $x = 1.2$

$$f(x) := \frac{x+1}{x^2+1} \quad f(1.2) = 0.902$$

Определим дискретные значения аргумента  $x \in [0, 10]$  с шагом 1: щелкнув по свободному месту в рабочем документе, введем с клавиатуры  $x := 0, 1 .. 10$ .

Щелкнув по свободному месту в рабочем документе, введем с клавиатуры  $f(x) =$

Под именем функции немедленно появится таблица значений функции:



### Построение декартова графика функции

Построим график функции  $f(t) = e^{-t^2}$

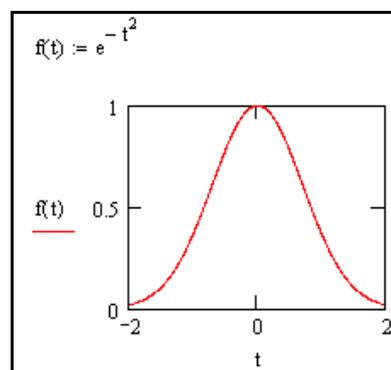
Щелкнем по свободному месту в рабочем документе и введем с клавиатуры  $f(t) := e^{-t^2}$

Щелкнем по свободному месту в рабочем документе, затем — по кнопке  в панели математических инструментов и в открывшейся панели щелкнем по кнопке .

Курсор установлен в помеченной позиции возле оси абсцисс. Введем с клавиатуры имя аргумента  $t$  затем щелкнем по помеченной позиции возле оси ординат, введем с клавиатуры  $f(t)$  и щелкнем вне прямоугольной рамки.

Определим промежуток изменения аргумента равным  $[-2, 2]$ . Для этого щелкнем по полю графика, затем — по числу, задающему наименьшее значение аргумента (число в левом нижнем углу ограниченного рамкой поля графиков), нажмем на клавишу  $\langle \text{Backspace} \rangle$  и введем с клавиатуры  $-2$ . Аналогично изменим вторую границу — вместо числа в правом нижнем углу поля графика введем  $2$ .

Щелкнем мышью вне поля графика:



## РЕШЕНИЕ ЗАДАЧ ЭЛЕМЕНТАРНОЙ МАТЕМАТИКИ В MATHCAD

### Преобразование алгебраических выражений

В MathCAD можно выполнить следующие символьные преобразования алгебраических выражений:

**simplify** (упростить) — выполнить арифметические операции, привести подобные, сократить дроби, использовать для упрощения основные тождества (формулы сокращенного умножения, тригонометрические тождества и т.п.);

**expand** (развернуть) — раскрыть скобки, перемножить и привести подобные;

**factor** (разложить на множители) — представить, если возможно, выражение в виде произведения простых сомножителей;

**solve** (решить) – решить уравнение.

Если MathCAD не может выполнить требуемую операцию, то он выводит в качестве результата вычислений исходное выражение.

Следует помнить, что MathCAD далеко не всегда преобразует выражение к самому простейшему виду.

**Пример 1.** Упростите выражение:  $\left(1 + \frac{2}{3 \cdot x - 1}\right) \cdot \left(1 - \frac{9 \cdot x - 9 \cdot x^2}{3 \cdot x + 1}\right) + 1$ .

Для того чтобы упростить введенное выражение, щелкнем по кнопке  в панели **Math** и в открывшейся панели щелкнем по кнопке **Simplify**, затем щелкнем мышью вне формулы. Результат (преобразованное выражение) будет отображен в рабочем документе справа от исходного выражения:

$$\left(1 + \frac{2}{3 \cdot x - 1}\right) \cdot \left(1 - \frac{9 \cdot x - 9 \cdot x^2}{3 \cdot x + 1}\right) + 1 \text{ simplify} \rightarrow 3 \cdot x$$

**Пример 2.** Раскройте скобки:  $(a + b)^3$

Для того чтобы раскрыть скобки, используем меню символьных операций. Щелкнем в панели операторов символьных вычислений по кнопке **expand**, затем щелкнем мышью вне формулы. Результат (преобразованное выражение) будет отображен в рабочем документе справа от исходного выражения:

$$(a + b)^3 \text{ expand} \rightarrow a^3 + 3 \cdot a^2 \cdot b + 3 \cdot a \cdot b^2 + b^3$$

**Пример 3.** Разложите число 24 на множители.

Щелкнем в панели операторов символьных вычислений по кнопке **expand**, затем щелкнем мышью вне формулы:

$$24 \text{ factor} \rightarrow 2^3 \cdot 3$$

### Символьное решение уравнений и систем уравнений

**Пример 4.** Найдите корни уравнения:  $x^2 + x - 6 = 0$

Щелкнем в панели операторов символьных вычислений по кнопке **solve** (в позиции справа от **solve** – имя переменной, относительно которой нужно решить уравнение), затем щелкнем мышью вне формулы:

$$x^2 + x - 6 \text{ solve, } x \rightarrow \begin{pmatrix} -3 \\ 2 \end{pmatrix}$$

$$\begin{aligned} 5 \cdot x - 3 \cdot y &= 1, \\ x + 11 \cdot y &= 6. \end{aligned}$$

**Пример 5.** Решите систему уравнений:

$$\begin{cases} \text{?} \\ \text{?} \\ \text{?} \end{cases}$$

Введите с клавиатуры ключевое слово **Given** (дано), затем правее и ниже ключевого слова — левую часть первого уравнения системы, далее — символьный знак равенства (нажмите на клавиатуре клавиши  $\langle \text{Ctrl} \rangle + \langle = \rangle$ ) и правую часть уравнения. Аналогично введите остальные два уравнения системы. Правее и ниже последнего уравнения системы введите имя функции **Find** (найти), перечислите в скобках имена искомых переменных, выделите **Find(x, y)**, щелкните по

кнопке  $\rightarrow$  в панели . Вычисленное решение системы будет отображено после щелчка мышью вне выделяющей рамки в рабочем документе справа от стрелки  $\rightarrow$  в виде матрицы, каждый столбец которой содержит одно из решений системы:

$$\begin{array}{l} \text{Given} \\ 5 \cdot x - 3 \cdot y = 1 \\ x + 11 \cdot y = 6 \\ \text{Find}(x, y) \rightarrow \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \end{pmatrix} \end{array}$$

## РЕШЕНИЕ ЗАДАЧ МАТЕМАТИЧЕСКОГО АНАЛИЗА В MATHCAD

Операторы математического анализа собраны в панели, которая открывается щелчком мыши по кнопке  в панели **Math**.

### Вычисление пределов

Для вычисления пределов предназначены кнопки  $\lim_{\rightarrow a}$ ,  $\lim_{\rightarrow a^+}$ ,  $\lim_{\rightarrow a^-}$ .

Для того чтобы вычислить предел, щелкните по свободному месту в рабочем документе, затем щелкните по нужной кнопке, введите с клавиатуры в помеченных позициях имя или выражение допредельной функции и предельной точки, выделите все выражение и щелкните по кнопке  $\rightarrow$  в панели символьных операций .

$$\lim_{x \rightarrow \infty} \frac{x^2 + 6}{2 \cdot x^2 + 5} \rightarrow$$

затем по щелкаем свободному месту документа

$$\lim_{x \rightarrow \infty} \frac{x^2 + 6}{2 \cdot x^2 + 5} \rightarrow \frac{1}{2}$$

### Дифференцирование

Чтобы найти производную, щелкните по свободному месту в рабочем документе, по кнопке в панели инструментов мат. анализа  $\frac{d}{dx}$  или для вычисления производных высших порядков  $\frac{d^n}{dx^n}$ , введите с клавиатуры в помеченных позициях имя или выражение функции и аргумента, заключите все выражение в выделяющую рамку и щелкните в панели  по кнопке  $\rightarrow$

$$\frac{d}{dx} x \cdot \cos(2 \cdot x) \rightarrow$$

затем по щелкаем свободному месту документа

$$\frac{d}{dx} x \cos(2 \cdot x) \rightarrow \cos(2 \cdot x) - 2 \cdot x \sin(2 \cdot x)$$

### Интегрирование

Для того чтобы найти неопределенный интеграл, щелкните по свободному месту в рабочем документе, затем щелкните по кнопке  $\int$ , введите с клавиатуры в помеченных позициях имя или выражение подынтегральной функции, выделите всё выражение и щелкните по кнопке  $\rightarrow$  в панели символьных операций 

$$\int \frac{1}{x^2 + x} dx \rightarrow$$

затем щелкаем свободному месту документа

$$\int \frac{1}{x^2 + x} dx \rightarrow \ln(x) - \ln(1 + x)$$

Аналогично вычисляется определенный интеграл (фрагмент документа MathCAD):

$$\int_0^{\frac{\pi}{2}} \frac{1}{1 + \cos x} dx \rightarrow$$

$$\int_0^{\frac{\pi}{2}} \frac{1}{1 + \cos x} dx \rightarrow \frac{1}{2} \cdot \frac{\pi}{1 + \cos x}$$

### РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОЙ АЛГЕБРЫ В MATHCAD

В задачах линейной алгебры практически всегда возникает необходимость выполнять операции с матрицами.

Панель операций с матрицами и векторами открывается щелчком по кнопке  в панели математических инструментов. Перечислим функции некоторых кнопок панели:

-  — определение размеров матрицы;
- $\times^{-1}$  — вычисление обратной матрицы;
- $|\times|$  — вычисление определителя матрицы;
- $\times^T$  — транспонирование матрицы;

Для того чтобы выполнить какую-либо операцию с помощью панели инструментов, нужно

выделить матрицу  $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$  и щелкнуть в панели по кнопке операции, затем поставить знак =

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = 0$$

Пример фрагмента документа MathCAD с матричными операциями:

$$A := \begin{pmatrix} 3 & 2 & 2 \\ 1 & 3 & 1 \\ 5 & 3 & 4 \end{pmatrix} \quad B := \begin{pmatrix} 2 & 3 & 2 \\ 1 & 2 & -3 \\ 3 & 4 & 1 \end{pmatrix}$$

$$A + B = \begin{pmatrix} 5 & 5 & 4 \\ 2 & 5 & -2 \\ 8 & 7 & 5 \end{pmatrix} \quad A \cdot B = \begin{pmatrix} 14 & 21 & 2 \\ 8 & 13 & -6 \\ 25 & 37 & 5 \end{pmatrix} \quad A^{-1} = \begin{pmatrix} 1.8 & -0.4 & -0.8 \\ 0.2 & 0.4 & -0.2 \\ -2.4 & 0.2 & 1.4 \end{pmatrix} \quad B^T = \begin{pmatrix} 2 & 1 & 3 \\ 3 & 2 & 4 \\ 2 & -3 & 1 \end{pmatrix}$$

Сумма                      Произведение                      Обратная матрица                      Транспонирование

$$|B| = -6$$

Определитель матрицы B

### Задания для самостоятельного выполнения:

1. Вычислить:

а)

$$\begin{pmatrix} 2 & 1 & 2 \\ 3 & -3 & 8 \\ 8 & -7 & 5 \\ 2 & 4 & 5 \end{pmatrix} \cdot \begin{pmatrix} -1 & 2 \\ 4 & 5 \\ -8 & -2 \end{pmatrix} + \begin{pmatrix} 2 & 3 & -9 & 8 \\ 0 & 1 & 1 & 9 \\ -2 & 5 & 1 & 0 \\ -6 & 3 & -4 & 4 \end{pmatrix} \cdot \begin{pmatrix} -3 & 9 \\ 2 & 4 \\ 1 & -4 \\ 6 & 3 \end{pmatrix}$$

б)

$$\begin{pmatrix} -5 & 0 & 3 \\ 4 & 1 & -1 \\ 2 & -3 & 2 \\ 1 & 5 & 3 \end{pmatrix} \cdot \begin{pmatrix} 3 & 0 \\ -2 & 1 \\ 4 & 3 \end{pmatrix} \cdot \begin{pmatrix} -2 \\ 3 \end{pmatrix}$$

2. Даны две матрицы:  $A = \begin{pmatrix} 2 & 3 & 2 \\ 1 & 2 & -3 \\ 3 & 4 & 1 \end{pmatrix}$  и  $B = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 5 & 3 \\ 3 & 4 & 2 \end{pmatrix}$ .

Найти:  $A^{-1}, A^T, |B|, B^2, F=6A-5B$ .

3. Вычислить определитель:  $D = \begin{pmatrix} 2 & 3 & 4 & -5 \\ 23 & 12 & -9 & 7 \\ 1 & 0 & 65 & -7 \\ 1 & 2 & 3 & -8 \end{pmatrix}$ .

4. Найти значения функции  $z(x) = \ln(x^2 + 2)$  на промежутке  $[-5, 5]$ , если  $x$  меняется с шагом равным 1.

5. Построить графики функций:

$$y(x) = \sin(x-5) \cdot \cos(x-5) \text{ на отрезке } [-10, 20];$$

$$f(x) = -5x^2 + 4x - 2 \text{ на отрезке } [-20, 20];$$

$$g(x) = \sqrt[3]{x-1} \text{ на отрезке } [-20, 20].$$

6. Упростить выражение:  $\left(1 + \frac{2}{3x-1}\right) \cdot \left(1 - \frac{9x-9x^2}{3x+1}\right) + 1 ; \frac{\left(\frac{1}{x-y} + \frac{3xy}{y^3-x^3}\right)}{\left(\frac{x^2+y^2}{x^2-y^2} - \frac{x+y}{2x-2y}\right)}$ .

7. Раскрыть скобки  $(a+b)^3, (a-b)^3, (x-2)(y+2)^2$ .

8. Разложить на множители  $54, (x^2-y^2), (x^3+y^3), (x^3-y^3)$ .

9. Решить уравнение  $\sqrt[3]{(x-2)^2} - \sqrt[3]{(x-3)^2} = 0, x^3 - 2x^2 - 3x = 0$ .

10. Решить систему уравнений по формулам Крамера

$$\begin{cases} x + 4y - 3z = -7, \\ x - 3y + 2z = 0, \\ 2x - 5y - z = -1. \end{cases}$$

$$\begin{cases} \cdot \\ \cdot \\ \cdot \\ \cdot \end{cases}$$

11. Решить систему уравнений:

$$\begin{cases} x^2 + 4y = 7, \\ 5y + xy = 5; \end{cases} \quad \begin{cases} x(z+1)^2 - 2z(x+z) = 0, \\ (1+x^2)\sqrt[4]{y-2} - 2x^2 = 0, \\ \sqrt{y-2}(z-2) + z = 0. \end{cases}$$

$$\begin{cases} \cdot \\ \cdot \\ \cdot \end{cases} \quad \begin{cases} \cdot \\ \cdot \\ \cdot \end{cases}$$

12. Вычислить пределы

$$\lim_{x \rightarrow 2} \frac{x^2 - 6x + 8}{x^2 - 8x + 12}, \quad \lim_{x \rightarrow \infty} \left(\frac{x^2 + 1}{x^2}\right)^{x^2 + 1}, \quad \lim_{x \rightarrow \infty} \sqrt{x^2 - 2x} - \sqrt{5x^2 + 4}$$

13. Вычислить производные первого порядка при  $x=2$  функций:

$$f(x) = x^2 + 2x, \quad g(x) = \sqrt{x^3 + 2x}.$$

14. Найти производные первого, второго, третьего порядка функции:

$$z(x) = \frac{1}{1+4x^2} + \arctg 2x.$$

15. Найти неопределенные интегралы:

$$\int tg^2 x dx , \int x \sqrt{4-x^2} dx , \int \frac{dx}{\sqrt{1-2x-5x^2}} .$$

16. Вычислить определенные интегралы:

$$\int_1^e x \ln x dx , \int_0^1 \frac{xdx}{\sqrt{1+x^2}} , \int_0^\pi \sin 9x \cos 3x dx .$$

## Delphi

**Тема 1: элементы языка Object Pascal, типы данных, Математические выражения, функции и операции.**

### Элементы языка Object Pascal

#### *Основные понятия языка*

**Алфавит** языка Object Pascal включает буквы, цифры, специальные символы, пробелы, зарезервированные (ключевые слова).

**Цифры** – арабские цифры от 0 до 9.

**Ключевые (зарезервированные) слова** – это последовательность символов, имеющая в данном языке особый установленный смысл. Ключевые слова нельзя использовать в качестве идентификаторов. Ключевые слова выделяются жирным шрифтом.

Например, в Object Pascal имеются следующие зарезервированные слова:

<b>And</b>	<b>Do</b>	<b>Implementatio</b>	<b>Of</b>	<b>String</b>	<b>While</b>
<b>Array</b>	<b>Downto n</b>		<b>Or</b>	<b>Then</b>	<b>With</b>
<b>Begin</b>	<b>Else</b>	<b>In</b>	<b>Out</b>	<b>To</b>	<b>xor</b>
<b>Class</b>	<b>End</b>	<b>Initialization</b>	<b>Procedure</b>	<b>Type</b>	
<b>Const</b>	<b>for</b>	<b>Interface</b>	<b>Program</b>	<b>Until</b>	
<b>Construction</b>	<b>File</b>	<b>Mod</b>	<b>Record</b>	<b>Unit</b>	
<b>Destruction</b>	<b>Function</b>	<b>Nil</b>	<b>Repeat</b>	<b>Uses</b>	
<b>Div</b>	<b>If</b>	<b>Not</b>	<b>Set</b>	<b>Var</b>	

#### **Object**

**Идентификаторы** - это имена констант, переменных, типов, модулей, объектов и т.п.

Идентификаторы могут иметь различную длину, но значащими являются только первые 63 символа. Идентификаторы всегда начинаются буквой, за которой могут следовать буквы и цифры, не содержат пробелов и специальных символов.

**Оператор** – минимальная структура в программе, производящая законченное действие.

Оператор содержит ключевое слово, которое определяет его смысл.

#### *Данные. Структура типов данных. Простые типы данных*

Данные в Object Pascal бывают:

константы (сохраняют свое значение в течение работы всей программы);  
переменные (могут изменять значение в результате работы программы);  
значение функций или выражений.

Любые данные характеризуются своими типами. Тип определяет:  
 множество допустимых значений, которые может иметь то или иное данное;  
 множество допустимых операций, которые применимы к данному;  
 объем памяти, выделяемый для хранения данного.

Object Pascal характеризуется разветвленной структурой типов данных.

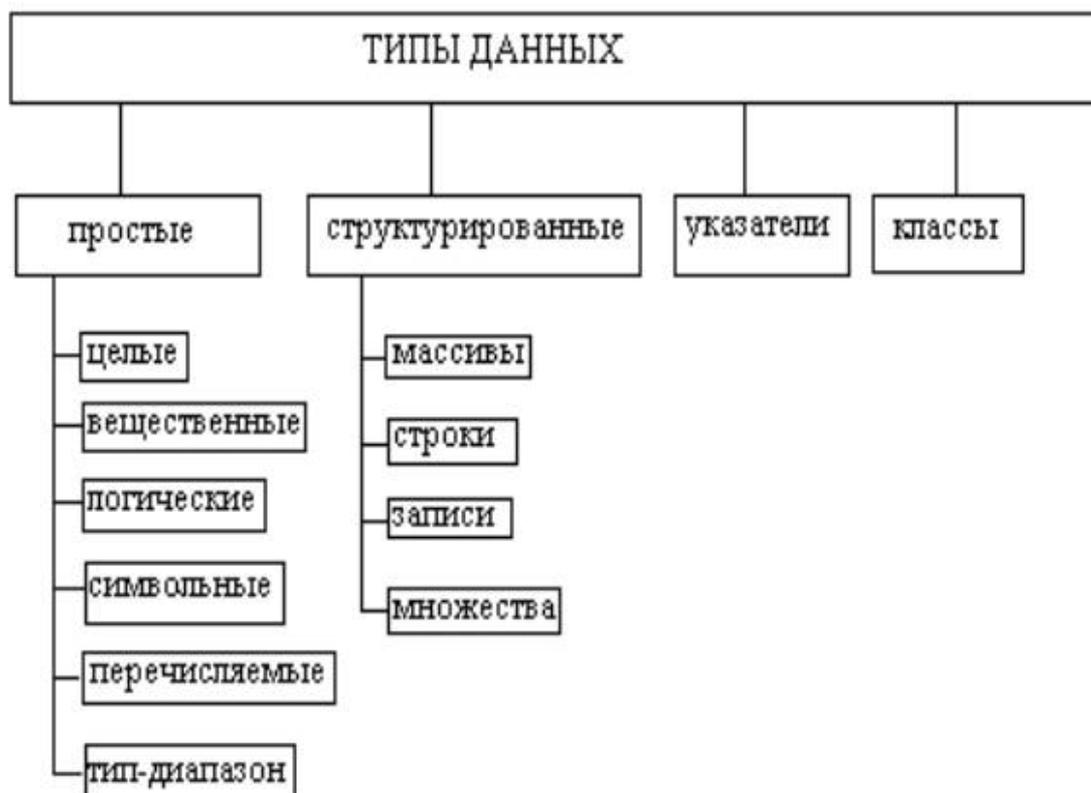


Рис.8.1.

### Типы данных Object Pascal.

Данные простого типа хранят в себе только одно значение. К простым типам относятся целые, вещественные, логические, символьные, перечисляемые, тип-диапазон.

**Целые типы** данных используются для представления целых чисел. Есть несколько различных целых типов, которые могут хранить целые значения. Различные целые типы имеют существенно различные диапазоны хранимых значений, затраты памяти растут с ростом допустимого диапазона значений.

Таблица 1. Целые типы данных.

Тип	Диапазон значений	Требования к памяти
Byte	[0; 255]	1 байт
Word	[0; 65535]	2 байт
Shorting	[-128; 127]	1 байт

Smallint	[-32768; 32767]	2 байт
Integer	[-2147483648; 2147483647]	4 байт
Cardinal	[0; 2147483647]	4 байт
Longint	[-2147483648; 2147483647]	4байт

**Вещественные типы** данных предназначены для хранения чисел, имеющих дробную часть.

В табл. 2 представлены несколько различных вещественных типов данных.

Таблица 2. Вещественные типы данных.

Тип	Диапазон значений	Требования к памяти
Real	$[2.9 \cdot 10^{-39}; 1.7 \cdot 10^{38}]$	6 байт
Single	$[1.5 \cdot 10^{-45}; 3.4 \cdot 10^{38}]$	4 байт
Extended	$[3.4 \cdot 10^{-4932}; 1.1 \cdot 10^{4392}]$	10 байт

**Логический тип данных** – один из простейших и часто используемых. Примером логического типа является **Boolean**. Переменные типа Boolean представляют собой логические значения True или False.

**Символьный тип данных.** Значениями символьного типа являются множество всех символов компьютера (буквы, цифры, арифметические знаки, орфографические знаки, специальные символы). Каждому символу приписывается код – целое число в диапазоне [0;255]. Код необходим для внутреннего представления символа.

Для кодировки в Windows используется код ANSI (назван по имени American National Standard Institute – Американского института стандартизации).

Примером символьного типа данных является тип **Char**, использующийся для хранения одного символа и в памяти отводится 1 байт.

### **Математические выражения, функции и операции**

**Математические выражения** – это операции с числами, переменными, значениями функций.

Порядок выполнения операций – слева направо, при этом сначала выполняются действия в скобках, после умножение, деление, арифметические функции, а дальше сложение, вычитание. Числа записываются при помощи цифр, причем целая часть от дробной отделяется точкой.

Например:  $4.5 \cdot (b - 2.2) / (c + 5.1)$ .

Числа могут быть представлены в показательной форме в виде:

1.76E-3 – запись соответствует  $1.76 \cdot 10^{-3} = 0.00176$

3.765E4 – запись соответствует  $3.765 \cdot 10^4 = 37650$

Основные арифметические операции и функции, используемые при записи выражений, представлены в табл. 3.

Таблица 3. Основные арифметические операции и функции.

Опера ция	Действие	Тип операндов	Тип результата	Пример
+	Сложение	Любой	Соответствует типу операндов	$4+5 = 9$
-	Вычитание			$9 - 4 = 5$
*	Умножение			$2 * 2 = 4$
/	Деление	Любой	Вещественный	$10/4 = 2.5$
Div	Целочисленное деление	Целый	Целый	$9 \text{ div } 4 = 2$
Mod	Остаток от деления	Целый	Целый	$9 \text{ mod } 4 = 1$
Abs	Модуль числа	Любой	Соответствует типу операнда	$\text{Abs}(-5.3)=5.3$
SQR	Возведение в квадрат	Любой	Соответствует типу операнда	$\text{SQR}(4)=16$
SQRT	Квадратный корень	Любой	Вещественный	$\text{SQRT}(78)=2.97$
Cos	Косинус	Любой	Вещественный	$\text{Cos}(4)=0.997$
Sin	Синус	Любой	Вещественный	$\text{Sin}(4)=0.069$
Exp	Экспонента	Любой	Вещественный	$\text{Exp}(4)$
Ln	Натуральный логарифм	Любой	Вещественный	$\text{Ln}(4)=1.38$
Round	Округление до ближайшего целого	Веществен- ный	Целый	$\text{Round}(4.5)=5$
Trunc	Возвращает целое число путем отбрасывания дробной части	Веществен- ный	Целый	$\text{Trunc}(4.5)=4$

### *Оператор присваивания*

Это наиболее часто используемый оператор языка. Синтаксис оператора присваивания:

Идентификатор:= выражение;

Символы «:=» всегда пишутся слитно, без разделяющих пробелов, хотя перед двоеточием и после знака равенства можно для лучшей читаемости программы вставлять пробелы. Как и любой другой оператор языка, оператор присваивания завершается точкой с запятой. Переменные и результат выражения должны быть одного типа.

### *Составной оператор*

Это последовательность произвольных операторов программы, заключенная в операторные скобки – **begin ... end**.

Составной оператор может состоять из любого количества операторов:

**begin**

оператор1;

оператор2;

...

операторN;

**end;**

Среди вложенных операторов могут быть и другие составляющие операторы:

**Begin**

.....

**Begin**

.....

**begin**

.....

.....

**end;**

.....

**End;**

.....

**end;**

## Тема 2: Знакомство с системой программирования Delphi

### Лабораторная работа №1

#### Цель занятия

**узнать:**

- два вида интерфейса MDI и SDI;
- структуру интерфейса Delphi;
- свойства: Font, Color, Visible, Name, Top, Width, Height, Left, Caption;
- событие OnClick;
- компоненты TLabel, TPanel, TButton;
- правило задания названия процедуры - реакции на события.

**научиться:**

- размещать компоненты на форме;
- задавать свойства и события объекта;
- описывать реакцию на событие в виде процедуры;
- сохранять и открывать программу: файлы \*.dpr, \*.pas, \*.dfm, \*.exe;
- запускать программу на исполнение и прекращать выполнение программы.

**Задача:** Программа «Светофор» для изменения цвета надписи.

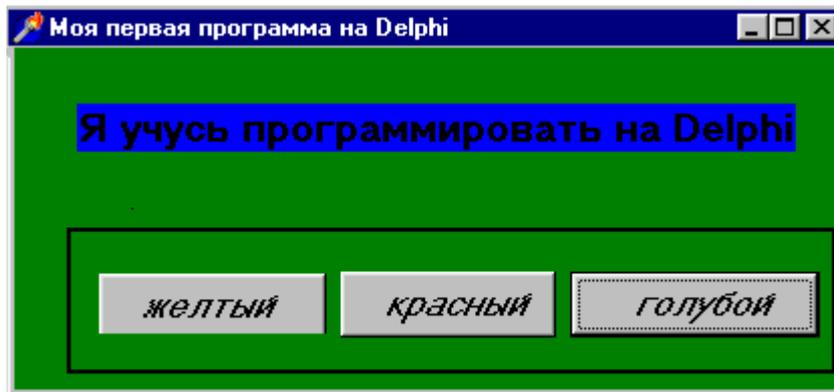


Рис.1.1. Пример окна программы «Светофор».

### Этапы выполнения задачи

#### **Этап 1. Структура среды программирования**

Внешний вид системы программирования Delphi отличается от многих приложений Windows. Все программы Microsoft Office (Word, Excel и др.) являются MDI приложениями. MDI (Multiple Document Interface) определяет особый способ управления несколькими дочерними окнами внутри одного большого окна. Delphi следует другой спецификации, называемой Single Document Interface (SDI) и состоит из нескольких отдельно расположенных окон. SDI близок интерфейсу, используемому в Windows. Окна могут перемещаться по экрану, частично или полностью перекрывая друг друга.

Интерфейс Delphi (рис. 1.2) представлен следующими окнами:

1. главное окно (Project1);
2. окно формы (Form1);
3. окно Инспектора Объектов (Object Inspector);
4. окно кода программы (Unit1.pas).

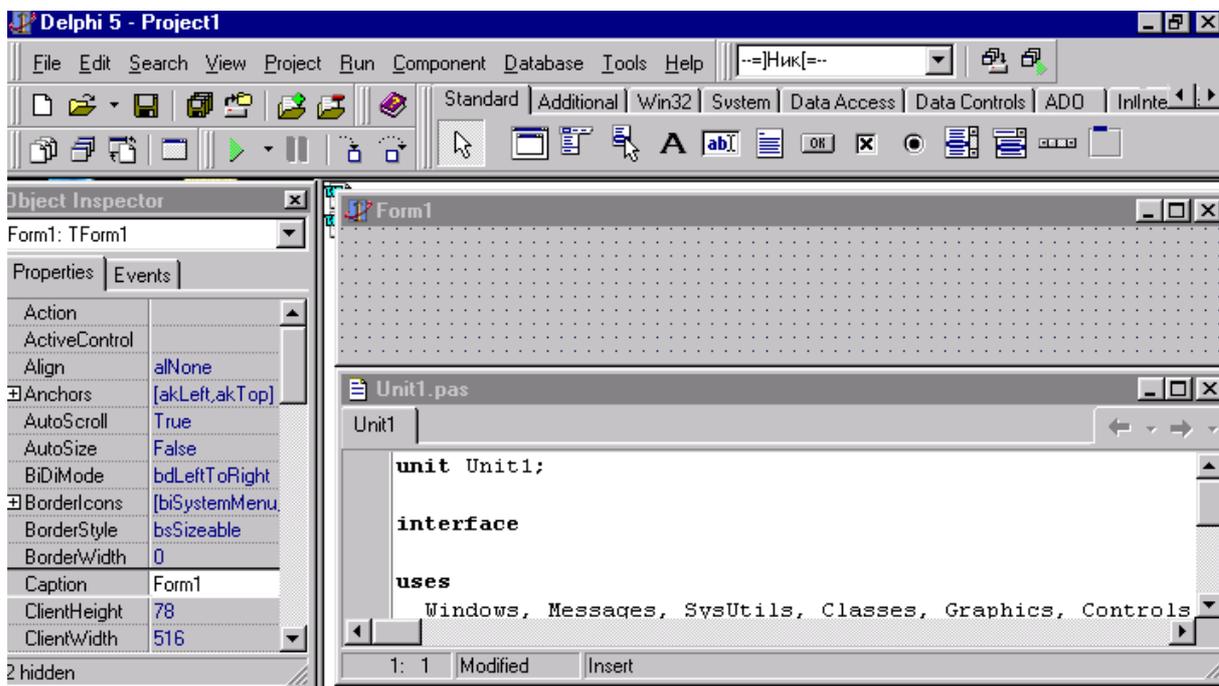


Рис. 1.2. Основные окна Delphi.

Главное (горизонтальное) окно и располагается в верхней части экрана. Оно содержит строку меню, панель инструментов и палитру компонентов. Палитра компонентов представлена группой отдельных страниц, каждая из которых снабжена закладкой. При подведении курсора мыши к компоненту появляется подсказка - название компонента.

Первоначально окно кода перекрыто окном формы, переключение между которыми осуществляется клавишей **F12**. Чтобы упорядочить окна так, как они показаны на рисунке, их размеры и местоположение можно изменить вручную. Расположение окон может быть произвольное, их расположение не влияет на работу Delphi.

### Задание 1.

- 1.1. Создайте папку на диске C: для сохранения файлов программы.
- 1.2. Запустите систему программирования Delphi.
- 1.3. Расположите окна Delphi согласно рис. 1.2, найдите каждое окно.

### Этап 2. Форма. Инспектор Объектов

**Форма** - окно будущей программы. Окно формы представляет собой проект окна Windows: имеет заголовок и системные кнопки. На форме размещаются компоненты, тем самым, формируя окно будущей программы.

Форма относится к классу TForm. Конкретное окно в программе является реализацией класса - объектом Form1. Имя объекта формируется путем задания название компонента (Form) и его порядкового номера (1). То есть, вторая форма получит имя Form2, далее – Form3 и т.д.

**Окно Инспектора Объектов** служит для задания свойств и событий объекта *Properties* (*Свойства*) и *Events* (*События*).

Основными **свойствами** компонентов являются:

Name - имя переменной, задаются по умолчанию. Имя объекта можно изменить, используя только английские буквы и цифры.

**Запомните!** Свойство Name обязательно, это имя объекта, используемое в программе.

Font - шрифт, позволяет выбрать параметры шрифта.

Caption – текстовая надпись, заголовок.

Visible - видимость, принимает значение True и False.

COLOR – цвет объекта, выбирается при помощи меню.

Width - ширина объекта.

Height - высота объекта.

**Задание 2:**

2.1. Для формы задайте свойства: имя - F1, размер - 500\*300, цвет - зеленый, шрифт - жирный курсив, заголовок (caption) - "Моя первая программа".

2.2. В окне кода программы посмотрите описание объекта формы.

### **ЭТАП 3. Компоненты TLabel, TButton, TPanel**

Для размещения компонента на форме нужно среди палитры компонентов щелкнуть мышью на данном компоненте, после щелкнуть на форме. Компонент можно удалить при помощи клавиши Delete.

**Компонент TLabel** - метка, служит для отображения текста на экране. Если цвет метки совпадает с цветом фона, то при сокращении надписи до нулевой длины, она может исчезнуть с экрана. В этом случае разыскать ее можно, перейдя в окно Инспектора Объектов, развернув список компонент и выбрав ее из этого списка.

**Компонент TButton** - командная кнопка. TButton позволяет выполнить какие-либо действия при нажатии кнопки во время выполнения программы.

**Компонент TPanel** - контейнер общего назначения. Панель имеет свойства:

BevelOuter (внешняя рамка) =  
BevelInner (внутренняя рамка) =  
i}i  
bvRaised (выпуклая),  
bvRaised (вдавленная),  
bvNone (отсутствует).

- i  
righ  
i  
i  
i  
[i][i]i  
iii

- BevelWidth (ширина рамки).

**Запомните!** Определяя свойства компонента, его необходимо выделить.

### Задание 3.

3.1. Разместите на форме TLabel. По умолчанию свойство name=Label1. Укажите надпись "Я учусь программировать на Delphi". Задайте жирный шрифт 14 размера. Выберите для метки, задаваемые в программе цвета: желтый, красный, голубой, выпишите названия цветов в тетрадь. Установите серый цвет метки.

3.2. Расположите ниже компонент TPanel. Задайте внешнюю и внутреннюю рамки, ширину рамки так, чтобы панель казалась вдавленной.

3.3. На панели разместите три командные кнопки с именами Button1, Button2, Button3 и надписями "Желтый", "Красный", "Голубой".

3.4. Откройте в окне Инспектора объектов список объектов, размещенных на форме. Проверьте наличие каждого компонента на форме.

3.5. В окне кода программы найдите созданное описание всех компонентов.

### Этап 4. События. Реакция на события. Окно кода программы

**События** устанавливаются в окне Инспектора Объектов на странице Events (События). Наиболее часто используемым событием Windows является *OnClick* - событие щелчок мыши.

В нашей программе при нажатии мыши (событие OnClick) на командные кнопки будет соответственно изменяться цвет метки.

Для задания события OnClick нужно два раза щелкнуть мышью на компоненте. После выбора события автоматически открывается окно кода программы.

**Окно кода программы** предназначено для создания и редактирования текста программы на языке программирования. В системе программирования Delphi используется язык программирования Object Pascal.

После задания события в окне кода программы автоматически создается процедура обработки события. Заголовок процедуры формируется:

**Procedure** тип формы. Имя объектаСобытие ();

**Procedure** TF1. Button1Click (Sender:TObject); {событие нажатие мыши на Button1}

**Begin**

**End;**

Между Begin End указывается **реакция на события - перечень совершаемых действий**.

При нажатии на командные кнопки Button1 ("желтый"), Button2 («красный»), Button3 («голубой») цвет метки должен изменяться соответственно.

В этом случае в процедурах описываются следующие действия:

**Procedure** TF1. Button1Click (..);

**Begin**

Label1.color:=clyellow;

**End;**

**Procedure** TF1. Button2Click (..);

**Begin**

Label1.color:=\_\_\_\_\_;

**End;**

**Procedure** TF1. Button3Click (..);

**Begin**

Label1.color:=\_\_\_\_\_;

**End;**

**Задание 4.** Создайте поочередно события и реакцию на события для командных кнопок. Названия цветов были выписаны на этапе 3.

### Этап 5. Отладка, запуск, сохранение программы

Сохранение программы осуществляется в два этапа:

- **File/Save as ...**- сохраняет модуль Pascal с расширением .pas. Имя файла задается английскими буквами.
- **File/Save project as ...**- сохраняет проект, основную программу, с расширением .dpr.

Имя модуля и проекта не должно совпадать. Поэтому удобно задавать имена файлов следующим образом:

- 1) имя модуля – lab1.pas;
- 2) имя проекта - P\_lab1.dpr.

При сохранении программы автоматически создаются несколько файлов:

\*.dfm - файл формы;

\*.exe - выполняемый файл, создается после запуска программы на исполнение.

**File/Open** – открытие файла программы. Запомните! Для открытия программы в Delphi выбирается файл проекта. При открытии файла проекта автоматически открываются файл модуля и формы.

**Run/Run** – запуск программы на исполнение. При запуске программы на исполнение программа компилируется – переводится в машинный код. Обнаруженные в процессе компиляции

синтаксические ошибки выделяются красным цветом, курсор помещается на символ, следующим за ошибкой. В строке состояния (нижней части окна) ошибка поясняется.

**Run/ Program Reset** - прекращение работы программы.

#### **Задание 5:**

- 5.1. Сохраните модуль и проект в папку, созданную на 1 этапе.
- 5.2. Запустите программу на исполнение, исправляя возникшие ошибки.
- 5.3. Прекратите выполнение программы. Закройте Delphi (главное окно).
- 5.6. Откройте Вашу папку и найдите созданные файлы (\*.pas, \*.dpr, \*.dfm, \*.exe). 5.7. Запустите выполняемый файл программы. Откройте файл проекта.

#### **Дополнительный этап: ОКНА СООБЩЕНИЯ**

Для открытия окна сообщения существует команда:

*ShowMessage ('текст сообщения');*

Например, мы хотим вывести сообщение о том, кто разрабатывал данную программу.

Создайте командную кнопку Button4 - «Автор». Процедура обработки события примет вид:

**Procedure** TF1.Button4Click (Sender:TObject);

**Begin**

ShowMessage (' программа разработана Ивановым Сергеем');

**End;**

**Задание 6.** Создайте кнопку «Автор», событие и реакцию на данное событие.

#### **Упражнения**

**1. Составьте программу, которая при выборе соответствующей командной кнопки, изменяет:**

*уровень начальный*

а) цвет **формы** на голубой, фиолетовый, серый;

*уровень средний*

б) текстовую надпись на "Язык программирования", "Система программирования", "Алгоритм";

в) размер формы на "крупный" (600\*800), "средний" (400\*600), "мелкий" (200\*400);

*уровень высокий*

г) вид панели "выпуклая", "вдавленная", "отсутствует";

д) видимость – невидимость текстовой надписи.

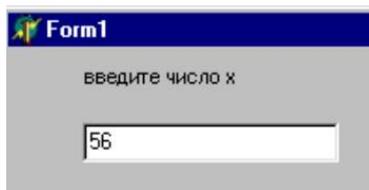
**Дополнительно. Создайте окно сообщения:**

а) об условии задачи;

б) о назначении компонентов при щелчке мыши на данном компоненте.

### Тема № 3: Программирование вычислительных задач, TEdit

С помощью компонента TEdit можно вводить или отображать достаточно длинные строки.



Центральным свойством компонента является свойство Text, которое представляет собой отображаемую строку.

#### Свойства компонента:

AutoSelect: Boolean	Указывает, будет ли выделяться весь текст в момент получения компонентом фокуса ввода.
AutoSize: Boolean	Если содержит True и BorderStyle=bsSingle, высота компонента автоматически меняется при изменении высоты шрифта.
TborderStyle = bsNone, bsSingle;	Определяет стиль оформления компонента: bsNone – нет оформления, bsSingle – компонент обрамляется одной линией.
MaxLength: Integer;	Определяет максимальную длину текстовой строки. Если имеет значение 0, длина строки не ограничена.
Modified: Boolean;	Содержит True, если текст был изменен.
ReadOnly: Boolean;	Если содержит True, текст не может изменяться.
SelLength: Integer;	Содержит длину выделенной части текста.
SelStart: Integer;	Содержит номер первого символа выделенной части текста.
SelText: String;	Содержит выделенную часть текста.
Text: String;	Содержит весь текст.

#### Методы компонента:

Clear;	Удаляет весь текст.
ClearSelection;	Удаляет выделенный текст.
CopyToClipboard;	Копирует выделенный текст в буфер обмена Clipboard.
CutToClipboard;	Копирует выделенный текст в буфер обмена Clipboard, после чего удаляет выделенный текст.
SetFocus;	Устанавливает фокус ввода.
PasteFromClipboard;	Вставляет текст из буфера обмена Clipboard.
SelectAll;	Выделяет весь текст.
Undo;	Восстанавливает текст в том виде, в каком он был перед последним получением компонентом фокуса ввода.

## Лабораторная работа №2

### Программирование вычислительных задач

#### Вопросы теоретического контроля

1. Понятие алгоритма. Линейная алгоритмическая структура. Блок-схема.
2. Понятие языка программирования: оператор, ключевые слова, идентификатор.
3. Типы данных Object Pascal: integer, real, string.
4. Функции перевода числа в строку, строку в число.
5. Математические функции.
6. Оператор присваивания.
7. Структура программных единиц: проекта, модуля, подпрограмм.

#### Цель занятия

узнать:

- этапы решения вычислительной задачи;
- свойства компонента TEdit;
- этапы выполнения вычислительных задач.

научиться:

- рисовать линейные блок-схемы;
- составлять простейшие программы вычисления;
- формировать структуры проекта.

**Задача:** Зависимость спроса  $y$  (ед.) в сутки на 1000 жителей от цены  $x$  (руб.) выражается степенной функцией\*:  $y=3,4*x^{0,4}$ . Составить программу оценки спроса на товар.

#### Этапы выполнения задачи

##### Этап 1. Анализ требований программы

В задаче вычисления значения выражения используются данные:

- *входная переменная:*  $x$
- *выходная переменная:*  $y$

Пусть переменная  $x$  будет целочисленна. Результат  $y=3,4*x^{0,4}$  будет вещественным числом.

Описание переменных примет вид:

- $x$ : *целое число;*
- $y$ : *вещественное число;*

**Задание 1.** Запишите задание и переменные программы в тетрадь.

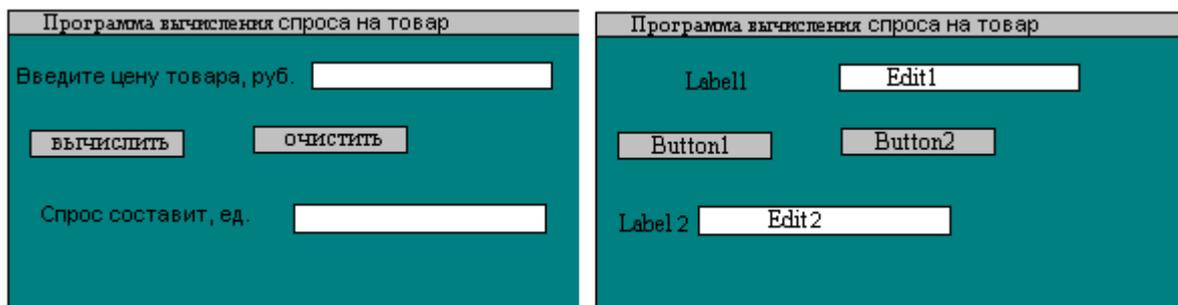
##### Этап 2. Окно будущей программы

*Для визуального проектирования* зарисовываются два окна. В первом окне определяется внешний облик окна, во втором окне указываются используемые элементы управления (в частности, компоненты).

Окно программы вычисления спроса удобно представлять:

---

\*



В окне исполняемой программы «ответ» первоначально должен быть невидим. Поэтому для компонентов вывода Label2 и Edit2 устанавливается свойство видимости `visible=false`. Напомним, что для всех компонентов по умолчанию свойство `visible=true` (компоненты видны на форме).

Свойства объектов формы:

**Форма:** `name=S1`, `caption=Вычисление спроса на товар`, `color` ....

**Метка:** `name=Label1`, `caption=Введите цену товара, руб.`, `font (шрифт) = 12 пт`

`name=Label2`, `caption=Спрос составит, ед.`, `font (шрифт) = 12 пт`, `visible=false`

**Командная кнопка** `name Button1`, `caption=вычислить`

`name Button2`, `caption=очистить`

**Однострочное окно** `name=Edit1`, `text` (пустая строка)

`name=Edit2`, `text` (пустая строка), `visible=false`

**Задание 2.** Создайте в Delphi окно будущей программы.

### Этап 3. Описание событий

После визуального проектирования окна формы определяются события программы. В программе будут происходить события:

- ✓ щелчок мыши на командной кнопке "Вычислить".
- ✓ ДОПОЛНИТЕЛЬНО: щелчок мыши на "Очистить".

Для каждого события в Delphi создаются процедуры:

**Procedure** `TS1.Button1Click()`; **Procedure** `TS1.Button2Click()`;

**Задание 3.** Создайте события программы.

### ЭТАП 4. Алгоритмизация и программирование

На этапе алгоритмизации составляется алгоритм каждого события. Алгоритмы реализуются на языке программирования.

Процедура «Вычислить» примет вид:

**Procedure** `TS1.Button1Click()`;

**Var**

`x:Integer`; `y: real`;

**begin**

**Заголовок** подпрограммы

**Переменные**

`x:целое`; `y:вещественное`;

```

x:=StrToInt(edit1.text);
y:=3.4*exp(0,4*ln(x));
label2.visible:=true;
edit2.visible:=true;
edit2.text:=FloatToStr(S);
end;

```



Алгоритм и процедура «Очистить» следующие:

<p><b>Заголовок подпрограммы</b></p> <pre> graph TD     Start([Начало]) --&gt; Clean1[очистить Edit1]     Clean1 --&gt; Clean2[Очистить Edit2]     Clean2 --&gt; Label[Сделать невидимым Label2]     Label --&gt; Edit[Сделать невидимым Edit 2]     Edit --&gt; Cursor[Курсор поместить в Edit1]     Cursor --&gt; End([Конец]) </pre>	<pre> <b>Procedure</b> TS1.Button2Click(); <b>begin</b> edit1.clear; edit2.clear; label2.visible:=false; edit2.visible:=false; edit1.SetFocus; <b>end;</b> </pre>
---	---

#### Дополнительный этап: Очистка компонентов формы

Очистка компонентов формы осуществляется следующим алгоритмом:

<pre> graph TD     Start([Начало]) --&gt; Clean1[очистить Edit1]     Clean1 --&gt; Clean2[Очистить Edit2]     Clean2 --&gt; Clean3[Очистить Edit3]     Clean3 --&gt; Label[Сделать невидимым Edit3]     Label --&gt; Cursor[Курсор поместить в Edit1]     Cursor --&gt; End([Конец]) </pre>	<pre> <b>Procedure</b> TS1.Button2Click(); <b>begin</b> edit1.clear; edit2.clear; edit3.clear; edit3.visible:=false; edit1.SetFocus; <b>end;</b> </pre>
---	---

Для программирования используются методы компонента TEdit:

**Clear** -удаляет весь текст.

**SetFocus** - устанавливает фокус ввода (курсор).

**Задание 4.** Зарисуйте алгоритм событий программы. Реализуйте процедуры.

#### ЭТАП 5. Тестирование и отладка программы

Напомним, тестирование – обнаружение ошибок, отладка – их исправление. Ошибки бывают синтаксические и структурные. Как уже говорилось, при запуске программы осуществляется компиляция, в процессе которой обнаруживаются синтаксические ошибки.

После запуска программы работающую программу проверяют на наличие структурных ошибок. Например, в нашем задании необходимо на бумаге задать значение x. При помощи

калькулятора посчитать значение выражения. Ввести эти данные в программу и проверить вычисленный результат с рассчитанным Вами.

Пусть цена товара  $x=50$  руб., тогда спрос  $y=3,4*50^{0,4} = 16$  (ед.).

**Задание 5.** Приведите свой пример проверки результатов программы.

### ЭТАП 6. Исполняемая программа

Конечный программный продукт должен быть подготовлен к реализации – в нашем случае к демонстрации студенческой группе и преподавателю. Для этого:

- Необходимо сохранить программу. Например, сохраним модуль с именем *Sproc.pas*. Файл проекта удобно формировать путем добавления начальной буквы *P* – *Pspros.dpr*
- Убедиться, что файл модуля прикреплен к проекту. Команды *Project/View source*, *Project/Add to project*, *Project/Remove from project*.
- Желательно подготовить сопроводительную информацию: ФИО разработчика, формулировка задачи и др. Например, *сопровождение программы лаб./работы №1*.

**Структура модуля** в окне кода программы примет вид:

```
unit Sproc; {имя модуля совпадает с именем файла Sproc.pas}
interface
{список модулей, с которыми устанавливается связь}
uses Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;
type Ts1 = class(TForm) {объявления типа TS1 класса формы}
    Label1: TLabel; {перечень компонентов формы типа TS1}
    Label2: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Button1: TButton;
    Button2: TButton;
procedure Button1Click(Sender: TObject); {объявления заглавия процедур}
private {переменные раздела private доступны в пределах данного модуля }
public {переменные раздела public доступны в пределах модуля и за его пределами}
    end; {завершение описания типа}
мфк ы1Ж Еы1ж чЖШтеупуж нЖ куфдж Хглобальные переменные модуляЪ
implementation {$R *.DFM}
procedure Ts1.Button1Click(Sender: TObject);
var x:Integer; y: real; {переменные доступны внутри процедуры – локальные переменные
begin подпрограммы}
x:=StrToInt(edit1.text);
y:=3.4*exp(0,4*ln(x));
label2.visible:=true; edit2.visible:=true;
```

```
edit2.text:=FloatToStr(S);
```

```
end; end.
```

Обратите внимание, что переменные можно описывать в разных разделах, в зависимости от требований программы.

Текст проекта (*Project/View source*) принимает вид:

```
program Pсproc;
```

```
uses Forms,
```

```
Sproc in 'Sproc.pas' {s1}; {подключили модуль Sproc и форму S1 файла 'Sproc.pas' }
```

```
{$R *.RES}
```

```
begin
```

```
Application.Initialize;
```

```
Application.CreateForm(Ts1, s1);
```

```
Application.Run;
```

```
end.
```

### Упражнения

1. Производственная функция Кобба-Дугласа имеет вид:  $P=5*L^{0,4}*K^{0,6}$ , где P-объем продукции, L-затраты труда, K-величина капитала. Составить программу вычисления объема продукции.
2. Объем V выработки молока фермерского хозяйства описывается моделью:  $V_{t0(1997)} = 5,6*e^{3(t-t0)}$ . Составить программу вычисления V в году t.
3. Спрос на получения кредита S (руб.) семьи в год для семьи со среднемесячный доход семьи на одного человека D (руб.) и количество детей в семье m описывается моделью:  $S = 3,4+5^m +4,3LnD$ . Составить программу вычисления спроса.
4. Для вклада S руб. под n% годовых рассчитать сумму через год.
5. Программа вычисления стоимости телефонного разговора. Запрашивает стоимость минуты и длительность разговора, а затем вычисляет стоимость разговора.
6. Для суммы S (руб.) делается скидка n%. Рассчитать скидку в руб. и сумму оплаты.

## Тема №4: операторы ветвления, TCheckBox

### Операторы ветвления

#### *Операции отношения. Логические операции*

Выражения можно сравнивать при помощи операции отношения:

= равно;

<> неравно;

< меньше;

> больше;

<= меньше или равно;

$\geq$  больше или равно.

Результатом применения операций отношения является логическое данное True или False.

Например, результатом выражения  $3 > 1$  является True, а результатом отношения  $3 < \diamond 3$  – False.

Сложные логические выражения составляются с использованием логических операций:

*Not* – логическое НЕ; *And* – логическое И; *Or* – логическое ИЛИ;  
*Xor* – исключительное ИЛИ.

Логические операции применимы к логическим данным и дают результат логического типа.

Можно привести следующий пример:

$X > 5$  and  $X \leq 10$ ,

где  $X > 5$  – операнд1 логического типа,  $X \leq 10$  – операнд2 логического типа, *and* – логическая операция. Результатом всего выражения будет логический тип.

Ниже в таблицах приведены правила формирования логического результата для выражения вида *операнд1 логическая операция операнд2*.

**Таблица 1.** Логическая операция **And** над данными логического типа.

Операнд1	Операнд2	Результат выражения <i>операнд1 and операнд2</i>
True	True	True
False	True	False
True	False	False
False	False	False

**Таблица 2.** Логическая операция **Or** над данными логического типа.

Операнд1	Операнд2	Результат выражения <i>операнд1 or операнд2</i>
True	True	True
False	True	True
True	False	True
False	False	False

**Таблица 3.** Логическая операция **Xor** над данными логического типа.

Операнд1	Операнд2	Результат выражения <i>операнд1 xor операнд2</i>
True	True	False
False	True	True
True	False	True
False	False	False

**Таблица 4.** Логическая операция **Not** над данным логического типа.

Операнд1	Результат выражения <i>not (операнд)</i>
False	True
True	False

*Условный оператор*

**Условный оператор (ветвление)** позволяет проверить некоторое условие и в зависимости от результатов проверки выполнять то или иное действие.

**Условие** в языке Object Pascal – это выражение логического типа (*BOOLEAN*), которое может принимать одно из двух значений: "истина" (*TRUE*) или "ложь" (*FALSE*). Условие записывается с помощью операций отношения и логических операций.

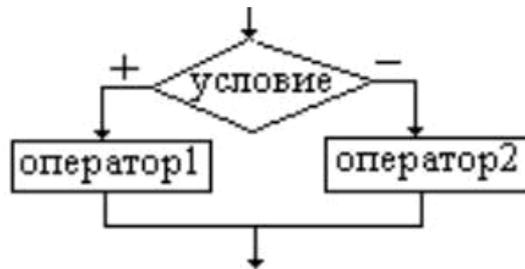
Существует два типа ветвления: *полное* и *неполное*.

**Структура полного ветвления** имеет следующий вид:

**If** <условие> **then** <оператор1> **else** <оператор 2>;

где **if, then, else** – ключевые слова (если, то, иначе).

Блок-схема полного ветвления имеет следующий вид:



Условный оператор работает по следующему алгоритму. Сначала вычисляется условие. Если результат условия *True* (истина), то выполняется оператор1, а оператор2 пропускается. Если результат условия *False* (ложь), то выполняется оператор2, а оператор1 пропускается. Например:

```
Var x, y, max: integer;
```

```
Begin
```

```
.....
```

```
  If x > max then y:= max else y:=x;
```

```
.....
```

```
End;
```

При выполнении этого фрагмента переменная *y* получит значение переменной *max*, если значение переменной *x* больше *max*; в противном случае *y* станет равно *max*.

Сложные логические выражения составляются с использованием логических операций **and, or, not, xor**. Например:

```
  If (x > max) and ( y>max) then y:= max else y:=x;
```

Часто в случае истинности или ложности условия выполняется более одного оператора, в таком случае используется составной оператор. Например, нахождение максимального из двух чисел *a* и *b* и вычисления их разности:

```
Var a,b,c: real; str: string;
```

```
Begin
```

```
.....
```

If a>b

Then begin

Str := 'a больше b';

C := a-b;

End

Else begin

Str := 'b больше a';

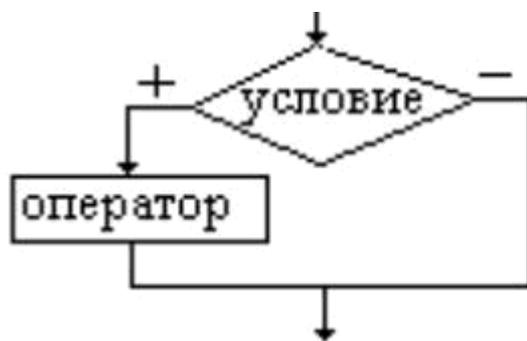
C := b-a;

End;

Структура неполного ветвления имеет следующий вид:

if условие then оператор;

Блок-схема неполного ветвления имеет следующий вид:



Неполный условный оператор работает по следующему алгоритму. Сначала вычисляется условие. Если условие принимает значение TRUE, то выполняется оператор, иначе никаких действий не происходит.

### TCheckBox - флажок



Флажок TCheckBox – независимый переключатель используется для того, чтобы пользователь мог указать свое решение типа *ДА/ НЕТ/ НЕ ЗНАЮ* (в последнем случае в окошке компонента устанавливается флаг выбора, но само окошко закрашивается серым цветом)

### Свойства компонента

Alignment = (taLeftJustify, taRightJustify);	Определяет положение текста с левой/ правой стороны компонента.
State = (cbUnchecked, cbChecked, cbgrayed);	Содержит состояние компонента cbUnchecked - нет, cbChecked - да, cbgrayed – не знаю.
AllowGrayed: Boolean;	Разрешает/ запрещает использование состояния cbGrayed ( <i>Не знаю</i> ).
Caption: String;	Содержит связанный с компонентом текст.
Checked: Boolean;	Содержит выбор пользователя True - ДА (cbChecked),

	False - НЕТ(cbUnchecked cbUnchecked, cbGrayed).
State = (cbUnchecked, cbChecked, cbgrayed);	Содержит состояние компонента cbUnchecked – нет, cbChecked – да, cbGrayed – не знаю.

Для использования компонента с возможностью выбора только ДА/НЕТ нужно свойство AllowGrayed = false (запрет значения НЕ ЗНАЮ), и в окне кода программы чаще всего имеется запись:

```
If checkBox1.Checked
    then .....
    else .....
```

В случае возможности выбора ДА/НЕТ/НЕ ЗНАЮ в окне кода программы используется типичная запись:

```
case CheckBox1.State of
    cbChecked: .....;
    cbunchecked: .....;
    cbGrayed: .....;
end;
```

Состояние флажка не зависит от состояния остальных флажков, поэтому такие переключатели называются *независимыми*. Свойство *Color* – компонента фактически игнорируется, а связанный с переключателем текст указывается в свойстве *Caption*.

### Оператор выбора

Оператор выбора позволяет выбрать одно из нескольких возможных продолжений программы. Параметром, по которому осуществляется выбор, служит **ключ выбора** – выражение любого целого, символьного и логического типа.

Структура оператора выбора такова:

```
case <ключ выбора> of
    <константа выбора>: оператор 1;
    <константа выбора>: оператор 2;
    .....
    <константа выбора>: оператор N;
else оператор
end;
```

Оператор выбора работает следующим образом. Вычисляется значение выражения <ключ выбора>, затем полученное значение последовательно сравнивается с константами выбора из списка констант. Если значение выражения совпадает с константой из списка, то выполняется соответствующая этому списку последовательность операторов, и на этом выполнение

инструкции **case** завершается. Если значение выражения не совпадает ни с одной константой из всех списков, то выполняется последовательность операторов, идущих после **else**.

Константа выбора может содержать не одно, а несколько значений, разделенных запятой, или представлять диапазон значений, в котором начальное и конечное значения разделены двумя точками.

Например, школьник учится в N классе. В зависимости от класса необходимо определить школьную ступень образования (начальное, среднее, старшие классы).

```
Var N: BYTE; Str:string;
begin
.....
case N of
  1..4: str := 'начальная школа';
  5..9 : str := 'среднее звено';
  10,11: str := 'старшие классы';
  else str := 'неверно указан номер класса';
end;
```

Иногда веточка **else** в операторе выбора может отсутствовать.

### Лабораторная работа №3

#### Программы с ветвлением. Компонент TCheckBox

##### Вопросы теоретического контроля

1. Понятие ветвления
2. Типы ветвления
3. Блок-схема полного и неполного ветвления
4. Синтаксис оператора ветвления.
5. Математические операции для записи условия
6. Назначение компонента TCheckBox.

##### Цель занятия

узнать:

- правила записи сложных условий неравенств типа  $0 < x < 10$  и др.;
- возможные состояния флажка TCheckBox, его свойства;
- понятие независимый переключатель.

научиться:

- программировать задачи нахождения значения функции на промежутках;
- составлять программы нахождения максимального элемента;
- записывать условия проверки состояния флажка.

##### Задача 1. Программа вычисления значения функции на промежутках

$$y = \begin{cases} x^3 + 3x + 4, & \text{если } x \in [0;1], \\ (x^3 + 3x + 4)^2, & \text{если } x < 0, \\ -4, & \text{если } x > 1. \end{cases}$$

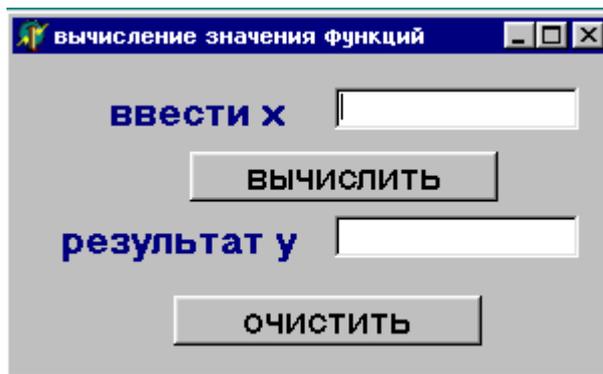


Рис. 3.1. Окно программы задачи 1.

### Этапы выполнения задачи 1

#### **Этап 1. Определение требований программы**

В программе используются две переменные: входная  $x$ , выходная  $y$ .

**Задание 1.** В тетради запишите условие задачи 1. На оси  $Ox$  отметьте промежутки, на которых функция описывается различными уравнениями.

#### **Этап 2. Окно будущей программы**

Для решения задачи форма формируется согласно рис.3.1. Для компонентов TForm, TEdit, TLabel задаются уже известные свойства.

Форма: name = Form1, Caption = Вычисление значения функции.

Метка: name = Label1, Caption = Введите X, Autosize=true.

Аналогично Label2.

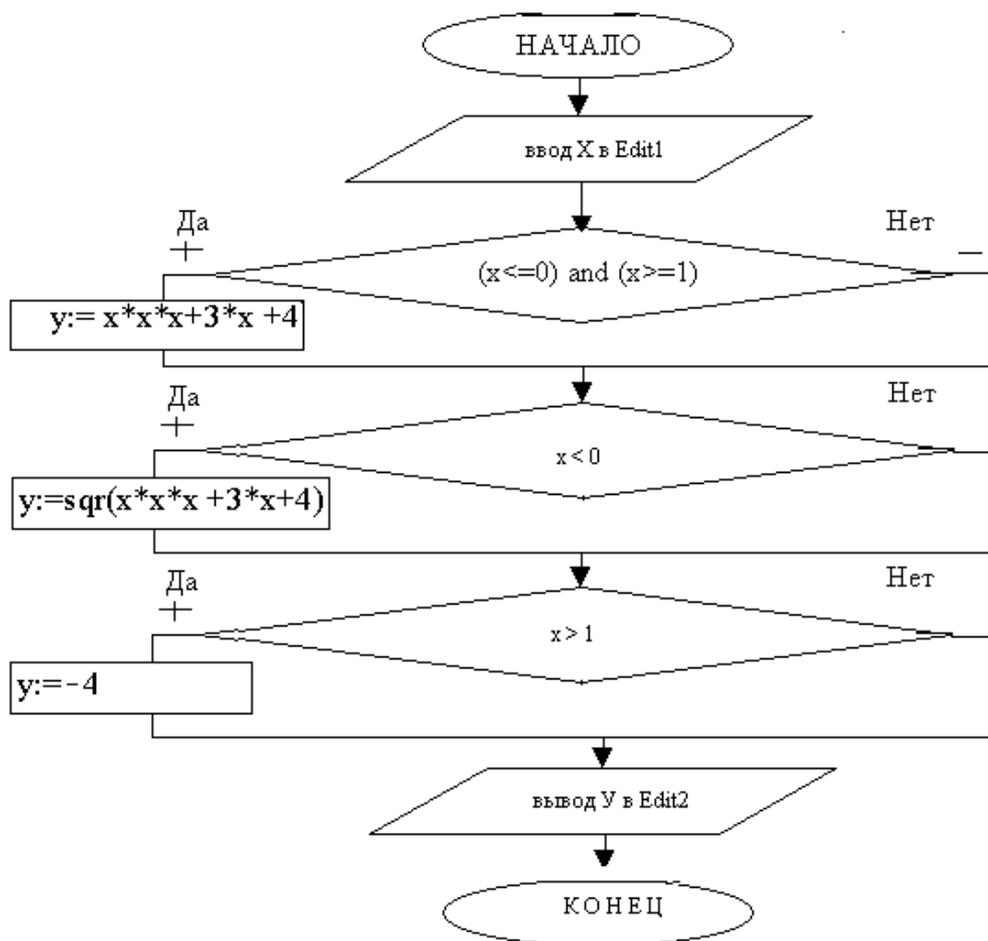
Однострочное окно: name = Edit1, Text €.

Аналогично Edit2.

**Задание 2.** В Delphi создайте окно будущей программы (окно формы).

### Этап 3. Алгоритмизация

Алгоритм **вычисления** значения функции определяет промежуток, к которому принадлежит введенное значение  $x$ .



В алгоритме проверяются условия:

1 условие:  $x < 0$

2 условие:  $x > 1$

3 условие:  $x \in [0; 1]$  - сложное, состоит из двух частей  $(x \leq 0)$  and  $(x \geq 1)$ .

Блок-схема решения задачи содержит три неполных ветвления проверки каждого условия.

Задание 3. Зарисуйте блок-схему в тетрадь. Выделите три блока ветвления.

### Этап 4. Программирование

В программе будут происходить два события:

1. Щелчок мыши по «Вычислить»: **Procedure** TForm1.Button1Click();
2. Дополнительно «Очистить»: **Procedure** TForm1.Button2Click();

Программирование алгоритма «Вычислить» записывается:

```
procedure TForm1.Button1Click(Sender: TObject);
    var x,y:real;
begin
```

```

x=strtfloat(edit1.text);
if (x>=0) and (x<=1) then y:= x*x*x+3*x +4;
if (x<0) then y:=sqr(x*x*x +3*x+4);
if (x>0) then y:=-4;
edit2.text:=floattostr(y);

```

**end;**

#### Задание 4.

4.1. Создайте событие "Вычислить" и запишите реакцию на него.

4.2. Дополнительно: Напишите реакцию на событие "Очистить" самостоятельно (см. лаб./работу №2).

### Этап 5. Тестирование

Наличие структурных ошибок проверяется тремя частными вариантами.

1) Ввести значение  $x \in [0;1]$ . Например, при  $x=0,5$   $y=x^3 + 3*x + 4=5,625$ .

2) Ввести значение  $x < 0$ . Например, при  $x=-2$   $y=(x^3 + 3*x + 4)^2=100$ .

3) Ввести значение  $x > 1$ . Например, при  $x=2$   $y=-4$ .

Расчетные результаты должны совпадать с результатами работающей программы. Несовпадение результатов показывает наличие структурных ошибок алгоритмизации или программирования.

**Задание 5.** Сохраните модуль и проект. Запустите программу на исполнение. Осуществите тестирование и отладку программы.

### Задача 2. Программа нахождения максимального значения

Max ( $5x^2 - 4y$ ;  $e^y$ ;  $\ln(x/2)$ )

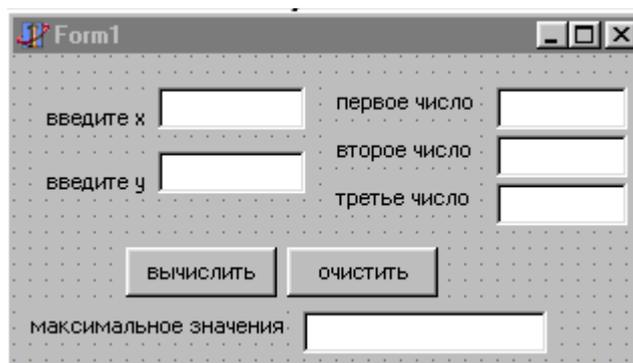


Рис. 3.2. Окно программы задания 2.

### Этапы выполнения задачи 2

#### Этап 1. Определение требований программы

Для нахождения  $\max(5x^2 - 4y; e^y; \ln(x/2))$  рассчитываются три значения:

- ✓  $a1 = (5x^2 - 4y)$  - первое значение;
- ✓  $a2 = e^y$  - второе значение;

✓  $a_3 = \ln(x/2)$  – третье значение,

среди которых находится максимальное.

Входными данными в задаче будут переменные  $x, y$ .

Промежуточными данными являются переменные  $a_1, a_2, a_3$ .

Выходом будет переменная  $\max$ .

**Задание 1.** Опишите в тетради входящие и выходящие данные.

### Этап 2. Окно формы

На рис. 3.3 показано расположение компонентов в окне формы.

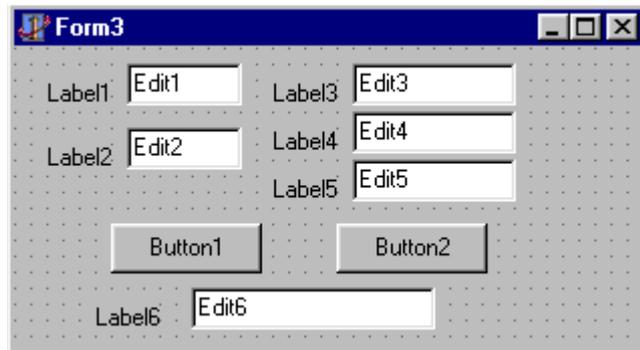


Рис. 3.3. Проектирование окна формы задачи 2.

В данной задаче не рассматривается задание свойств компонентов (подробно эти этапы рассматривалось в лабораторных работах №1, №2).

Усложним форму указанием видимости компонентов. Компоненты задания входных данных (Edit1, Edit2) сделаем видимыми при открытии формы: установим свойство `visible=true`. Компоненты отображения результатов (Edit3 – Edit6) первоначально невидимы: установим свойство `visible=false`.

**Задание 2.** В Delphi на оформите окно формы. Сохраните модуль и проект.

### Этап 3. События и реакция на события программы

В программе событие «Вычислить» реализуется процедурой:

```
procedure TForm1.Button1Click(Sender: TObject);  
    var x,y,a1,a2,a3, max: real;  
begin  
    x:=strtofloat(edit1.text);  
    y:= strtofloat(edit2.text);  
    a1:=5*x*x-4*y*y; Edit3.text:=floattostr(a1);  
    Edit3.visible:=true;  
    a2:=exp(y); Edit4.text:=floattostr(a2);  
    Edit4.visible:=true;  
    a3:=ln(x/2); Edit5.text:=floattostr(a3);  
    Edit5.visible:=true;
```

```

if (a1>a2) and (a1>a3) then max:=a1;
If (a2>a1) and (a2>a3) Then max:=a2;
IF( a3>a1) and (a3> a2) then max:=a3;
Edit6.text:=floattostr(max);
Edit6.visible:=true;

```

**end;**

Обратите внимание, что максимальное значение присваивается переменной max.

Типичная ошибка  $a1:=max$  *изменяет значение переменной a1* и нигде не записывает максимальное значение (max - неопределенно).

**Задание 3.** Создайте событие «Вычислить» и реакцию на него.

#### ЭТАП 4. Тестирование

Для тестирования программы решим задачу  $\max(5x^2-4y; e^y; \ln(x/2))$  при  $x=2, y=3$ . Для этого рассчитываются значения  $a_1, a_2, a_3$ , например, с помощью программу "Калькулятор" в Пуск/Программы/Стандартные. Среди  $a_1, a_2, a_3$  находится максимальное значение.

**Задание 4.** Сравните полученные Вами значения с результатом программы.

**Задание 5.** Алгоритмизация: Зарисуйте в тетради блок-схему решения задачи.

**Дополнительное задание.** Создайте кнопку «Условие задачи» с выводом окна (рис. 3.3).

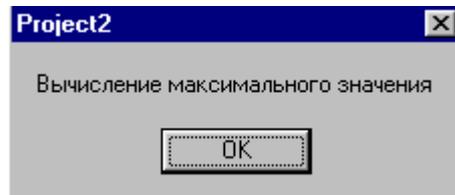


Рис. 3.3. Окно сообщение задачи 2.

#### Задача 3. Программа "Флажок"

Вычисление квадрата  
и модуля числа

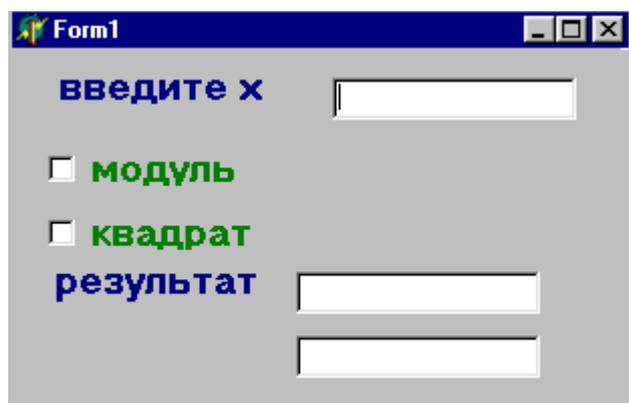


Рис. 3.4. Окно задачи 3.

### Этапы выполнения задачи 3

#### Этап 1. Определение требований программы

В рассматриваемой задаче:

- ✓ вход:  $x$  (целое число);
- ✓ выход:  $y_1$  (модуль числа  $x$ ) и  $y_2$  (квадрат числа  $x$ ).

**Задание 1.** В тетради опишите тип переменных  $x$ ,  $y_1$ ,  $y_2$ . Запишите функции нахождения модуля и квадрата числа.

#### Этап 2. Окно формы

В окне формы компоненты располагаются, как показано на рис. 3.5.

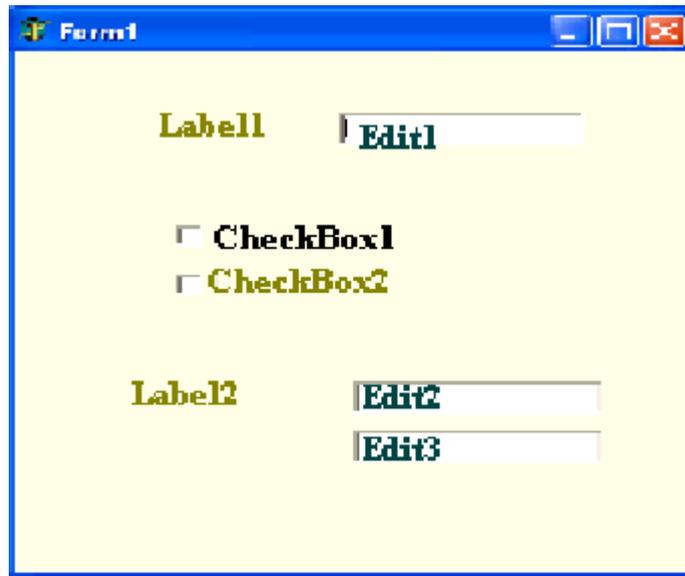


Рис. 3.5. Проектирование окна формы задачи 3.

Для компонентов TForm, TEdit, TLabel задаются уже известные Вам свойства.

Форма: name = Form1, Caption = Программа вычисления модуля и квадрата.

Метка: name = Label1, Caption = Введите X, Autocomplete = true.

Аналогично Label2, Label3.

Однострочное окно: name = Edit1, Text .

Аналогично Edit2, Edit3. Установите окна результата невидимыми.

#### **Флажок**

Кроме известных компонентов на форму помещается новый компонент флажок. Флажок может принимать три состояния:

- ДА – cbChecked;
- НЕТ – cbUnchecked;
- НЕ ЗНАЮ – cbGrayed.

Состояние флажка задается свойством **State** = [cbChecked, cbUnchecked, cbGrayed].

Работая в Windows, Вы встречались с флажком и замечали, что чаще всего флажок принимает только два состояния ДА-НЕТ. Свойство **AllowGrayed**=[true, false] включает/выключает третье состояние - НЕ ЗНАЮ.

## Задание 2.

- 2.1. Запишите в тетради свойства компонента TCheckBox: AllowGrayed и State.
- 2.2. Оформите окно формы Delphi.
- 2.3. Изучите состояния флажков. Поэкспериментируйте: для CheckBox1 задайте включен (ДА), для CheckBox2 - НЕ ЗНАЮ.
- 2.4. Установите для флажков свойства:

Флажок: name = CheckBox1 (CheckBox2)

Caption = модуль (квадрат)

State - флажки не включены.

### Этап 3. События изменения состояния флажков

Событие щелчок мыши по кнопке «модуль» - CheckBox1Click, находит абсолютное значение числа X, записанное в Edit1, и выводит его в Edit2.

**Procedure** TForm1.CheckBox1Click (Sender: TObject);

var x,y1: integer;

**begin**

x:=StrToInt(edit1.text);

y1:=abs(x);

edit2.text:= IntToStr(y1);

if checkbox1.state=cbChecked

then edit2.visible=true

else edit2.visible=false;

**end;**

Обратите внимание на условие для флажка: *checkbox1.state=cbChecked*. Условие истинно при включенном флажке. В этом случае пользователю выводится результат. При ложном условии флажок выключен, следовательно, результат не отображается на экране.

**Задание 3.** Создайте события для двух флажков, реакции на эти события.

### Этап 4. Тестирование и отладка

Введите отрицательное значение x, например, x=-2. Выберите флажок "модуль" и должны получить результат. Выключите флажок - результат исчезнет. Аналогично протестируйте флажок "квадрат".

Обратите внимание, Вы можете включить два флажка вместе, можете отключить все флажки, включать только один. Говорят, что состояние одного флажка не зависит от состояния других, поэтому **флажок - независимый переключатель**.

**Задание 4.** Выпишите в тетрадь понятия независимый переключатель. Создайте исполняемую программу. Не забудьте сохранить программу.

### Упражнения

1. Вычислить Y – размер ВВП страны в зависимости от:

a – индекса расходов на сельское хозяйство к бюджету страны (от 0 до 1)

b – индекса расходов на промышленность к бюджету страны (от 0 до 1)

z – индекс расходов на образование к общему бюджету страны (от 0 до 1)

x = a-b – разница индексов расходов на сельское хозяйство и промышленность

			<i>Уровень средний</i>
A	$100z^2, \text{ если } z \leq 0,3$ $120z^2 + 5 \ln z, \text{ если } 0,3 < z < 0,7$ $200z^3, \text{ если } z \geq 0,7$ $y = \begin{cases} 100z^2, & \text{если } z \leq 0,3 \\ 120z^2 + 5 \ln z, & \text{если } 0,3 < z < 0,7 \\ 200z^3, & \text{если } z \geq 0,7 \end{cases}$	E	$y = \begin{cases} a^2 + b^2, & \text{если } a^2 + b^2 \leq 1, \\ (a+b)^2, & \text{если } a^2 + b^2 > 1 \text{ и } a \geq b, \\ 0,5, & \text{если } a^2 + b^2 > 1 \text{ и } a < b. \end{cases}$
B	$y = \begin{cases} x^2, & \text{если } -0,5 \leq x \leq 0,5, \\ 4, & \text{если } x > 0,5, \\ \sqrt{-8x}, & \text{если } x < -0,5. \end{cases}$	F	$y = \begin{cases} 1/4, & \text{если } a = b \\ 2(a^2 + b^2), & \text{если } a > b \\ \operatorname{tg}(a+b), & \text{если } a < b \end{cases}$
C	$y = \begin{cases} 0, & \text{если } x \geq 1, \\ x, & \text{если } 0 < x \leq 1, \\ x^4, & \text{в противном случае.} \end{cases}$	G	$y = \begin{cases} 0, & \text{если } x \leq 0, \\ x^2 - x, & \text{если } 0 < x \leq 1, \\ x^2 - \sin \pi x^2 - 1, & \text{в противном случае.} \end{cases}$
D	$y = \begin{cases} x^2 + 4x + 5, & \text{если } x \leq 2, \\ \frac{1}{x^2 + 4x + 5}, & \text{в противном случае.} \end{cases}$		

*Уровень высокий.* Дополнить программу флажками. Задать три флажка, соответствующие каждому промежутку. При выводе результата включать соответствующий флажок.

2. Даны x, y, z. Составить программу нахождения максимального элемента

	<i>Уровень начальный</i>		<i>Уровень средний</i>
A	$\max(x+y+z, 3x, y-4z)$	F	$\min(\sin(x^2+y^2), \operatorname{tg}(z), x/y)$
B	$\min(x^2+y^2,  z , xyz)$	G	$\max(5x^2-4y^2), e^y, \ln x/2 $
C	$\max(x^2+y^2-z^2, xyz, (x-y)^4)$	I	$\max(\sin(x+y), \operatorname{tg}(10-2z), 6y^3)$
D	$\min( x+y , (z/x)^2, 45/x^3)$	J	$\max(\ln x-y , \operatorname{tg}(z), x/y)$
E	ышт(ышт(ч+н)б 5-6нб еп(я))	K	$\max(\sin(x-y), \operatorname{tg}^2(z), x/y)$

### Творческие задания

1. Напишите программу расчета скидки на покупку. Если сумма покупки составляет более 500 р., то покупателю предоставляется скидка n%.

2. Известна величина прожиточного минимума на человека в месяц. Напишите программу, которая для семьи  $n$  человек с доходом  $S$  включает флажок выше или ниже прожиточного минимума. Дополнительно: предусмотреть возможность равенства значений.
3. Некоторый товар можно приобрести в трех торговых точках. В каждом магазине известна цена товара  $S_1, S_2, S_3$  и предоставляемые скидки  $r_1, r_2, r_3$ . Найти минимальную цену с указанием магазина. Вывести реальную цену покупки в каждом магазине.
4. Стоимость печати стандартной визитки  $S$  рублей. При печати визитки на глянцевой бумаге стоимость увеличивается на 10%. Нанесение логотипа на визитку увеличивает стоимость печати на 5%. Печать более 100 экземпляров уменьшает стоимость на 8%. Написать программу расчета стоимости печати одной визитки. В программе выбор параметров осуществляется флажками «Глянцевая бумага», «Логотип», «Объем более 100 экз.».

#### Тема №4: Класс TStrings - набор строк

Абстрактный класс TStrings используется для работы с наборами строк. От него порождены потомки, обслуживающие наборы строк в таких компонентах как TComboBox, TListBox, TМemo и др.

Объекты TStrings обслуживают работу со строкой – объектом, поэтому строки можно сортировать, отыскивать нужный объект по описанию и т.д. Строки в наборе строки нумеруются (индексируются), начиная с нуля.

##### Свойства класса:

Commatext: string	Служит для получения всего набора строк в виде единой строки с кавычками и запятыми: «строка1», «строка2», «строка3», ...
Count: Integer;	Количество строк в наборе.
Strings[index:integer]:String;	Открывает доступ к строке с индексом Index.
Text:string;	Служит для получения всего набора строк в виде единой строки с разделителями EOLN: «строка1  строка2   строка3   строка4   ...»

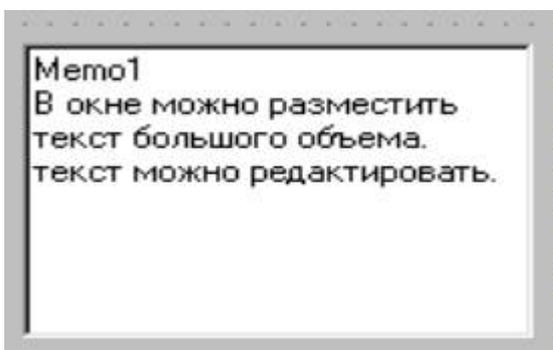
##### Методы класса:

Add(s:string):integer;	Добавляет строку в набор данных последней и возвращает ее индекс.
Append(s:string);	Добавляет строку в набор данных последней, но не возвращает ее индекс.
Clear;	Очищает набор данных.
Delete(Index:integer);	Удаляет строку с индексом Index.
Exchange(Index1, index2:integer);	Меняет местами строки с index1 и index2.
IndexOf(S:string):integer;	Определяет индекс строки S в списке или возвращает -1, если такой строки нет.

Insert(index:integer; s:string);	Вставляет строку в набор с индексом Index.
Loadfromfile(FileName:string);	Открывает файл и считывает набор строк.
SaveToFile(file:string);	Сохраняет набор строк в файл.
Move(CurIndex,NwIndex:integer);	Перемещает строку из положения CurIndex в положение NewIndex.

### ТМемо - многострочное окно ввода-вывода

Компоненты класса ТМемо предназначены для ввода, редактирования и отображения достаточно большого текста. Компонент представляет собой небольшой текстовый редактор, имеет ограничения на объем текста в 32Кб, что составляет 10-20 стр.



Текст хранится в свойстве Lines класса TStrings. В целом компонент представляет собой пронумерованный набор строк (нумерация начинается с нуля). С помощью свойств и методов класса TStrings (Count, Add, Delete, Clear и т.д.) можно динамически формировать содержимое компонента. Например:

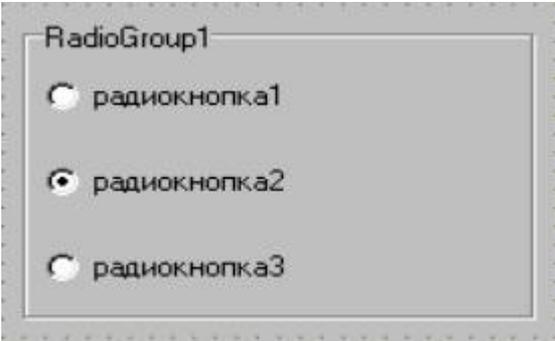
```
edit1.text:=memo1.Lines.Strings[0];
memo1.Lines.Insert(2,'Иванов');
memo1.Lines.SaveToFile('c:\Мои документы\prim1.txt');
```

Компоненту присущи специфичные свойства:

Lines: Tstrings;	Содержит строки текста.
TscrollStyle = (ssNone, ssHorizontal, ssVertical, ssBoth)	Определяет наличие полос прокруток.
WantReturns: Boolean	Если содержит True, нажатие Enter вызывает переход на новую строку.
WantTabs: Boolean;	Если содержит True, нажатие Tab вызывает ввод в текст символа табуляции, который ставится нажатием Ctrl+Tab.
ReadOnly: Boolean;	Если содержит True, текст не может изменяться.
MaxLength: Integer;	Определяет максимальную длину текстовой строки. Если имеет значение 0, длина строки не ограничена.

### TRadioGroup - группа радиокнопок

Компонент класса TRadioGroup представляет собой специальный контейнер для размещения радиокнопок (TRadioButton).



Каждый размещаемый на контейнере переключатель (радиокнопка) помещается в специальный список Items: TStrings. В данном списке каждая радиокнопка пронумерована, начиная с нуля. Работа со списком радиокнопок осуществляется аналогично работе с классом TStrings (см. пункт 12.4).

Свойства компонента:

Columns: Integer;	Определяет количество столбцов переключателей.
Items: TStrings;	Содержит список строк с заголовками радиокнопок. Добавление/удаление элементов достигается согласно строкам списка (см. п. 12.4).
ItemIndex: integer;	Содержит индекс включенной радиокнопки. Если индекс равен -1, то все радиокнопки выключены.

Типичная работа со списком радиокнопок:

```

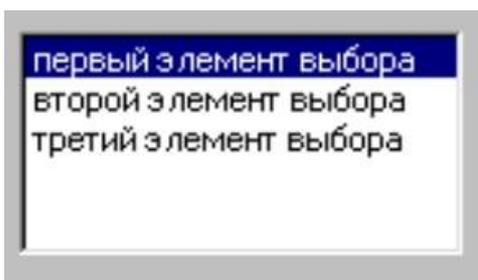
case radiogroup1.ItemIndex of
    0: .....;      {выбран 0 переключатель}
    1: .....;      {выбран 1 переключатель}
    2: .....;      {выбран 2 переключатель}
    .....
else .....      {не выбран ни один переключатель}
end;

```

### TListBox - список выбора

Компонент класса TListBox представляет собой стандартный список выбора, с помощью которого пользователь может выбрать один или несколько элементов. В компоненте

предусмотрена возможность программной прорисовки элементов, поэтому список может содержать не только строки, но и произвольные изображения.



Свойства компонента:

Canvas: Tcanvas;	Канва для программной прорисовки элементов.
Columns: Longint;	Определяет количество колонок элементов в списке.
ItemHeight: integer;	Определяет высоту элемента в пикселях.
ItemIndex: Integer;	Содержит индекс выбранного (выделенного) элемента.
Items: Tstrings;	Содержит набор строк элементов.
MultiSelect: Boolean;	Разрешает/отменяет выбор нескольких элементов.
SelCount: Integer;	Содержит количество выбранных элементов.
Sorted: Boolean;	Разрешает/отменяет сортировку строк в алфавитном порядке.

Элементы списка задаются в списке Items:TStrings, нумеруются с нуля. Создание элементов списка компонента реализуется с помощью методов свойства Items – Add, Append, Insert и других методов класса TStrings.

## Лабораторная работа №4

### Программы выбора из списка. Класс TStrings

#### Вопросы теоретического контроля

1. Понятие и синтаксис записи оператора выбора.
2. Принцип работы ключа выбора.
3. Компоненты класса TString: TRadioGroup, TMemo, TListBox.
4. Индексация строк компонентов класса TStrings.
5. Методы класса TString: Clear, SetFocus, Add, Delete, LoadFromFile, SaveToFile и др.

#### Цель занятия

УЗНАТЬ:

- правила записи констант выбора оператора CASE;
- представление набора строк в TListBox и TRadioGroup;
- возможности выбора радиокнопок в TRadioGroup, понятие зависимый переключатель;
- различия в использование флажка и радиокнопки;
- отличия компонентов TMemo и TListBox, TMemo и TEdit, TRadioGroup и TCheckBox, TListBox и TRadioGroup.

НАУЧИТЬСЯ:

- составлять программы с множественным выбором решения;
- использовать компоненты TRadioGroup и TListBox для выбора данных;
- выделять результат в списке через компоненты TRadioGroup и TListBox;
- выводить результат в TMemo.

**Задача 1: Программа выбора прессы в TListBox по дню недели**

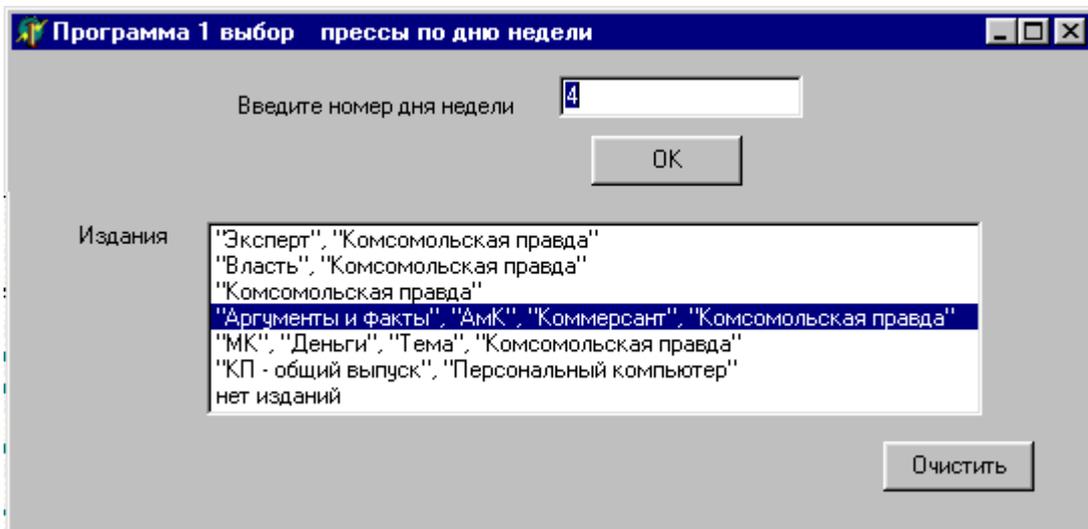


Рис.4.1. Пример окна программы «Периодические издания»-1.

**Этапы выполнения задачи 1**

**Этап 1. Определение требований программы**

В задаче 1:

✓ *входная переменная:*

n: целого типа (вводится в Edit1 - номер дня недели).

✓ *результат:*

действие - выделение строки в списке изданий TListBox.

**Задание 1.** В тетради опишите этап 1.

**Этап 2. Окно будущей программы**

Окно формы представлено на рис. 4.1.

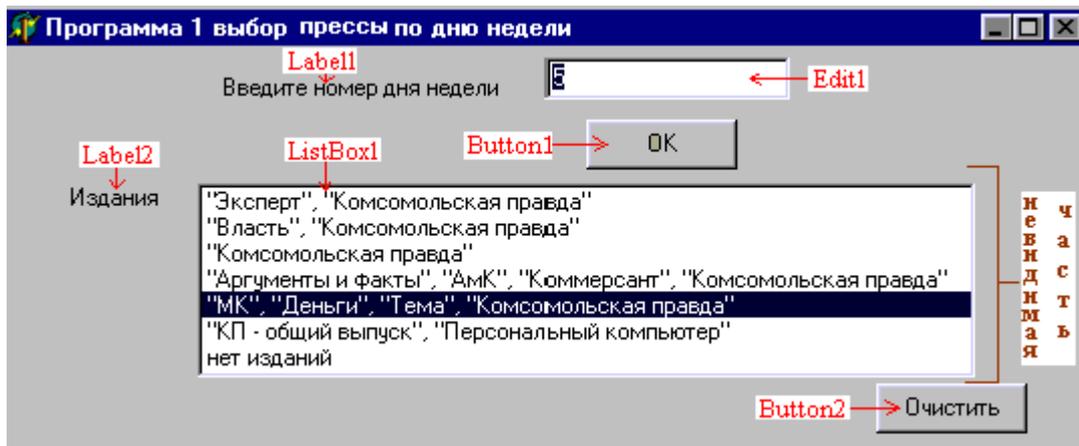


Рис. 4.2. Пример визуального проектирования окна программы 1.

В окне формы располагаются известные TEdit, TLabel, TButton. Зададим свойства этих компонентов.

Форма: name = F1, Caption = Программа 1 выбора прессы по дню недели.

Метки: name = Label1, Caption = Введите номер дня недели.

name = Label2, Caption = Издания в этот день.

Окно ввода: name = Edit1, Text .

Командная кнопка: name = Button1, Caption = ОК.

Разместим на форме новый компонент с именем ListBox1 в виде прямоугольного окна.

Размер компонента можно изменять при помощи мыши.

TListBox представляет список строк, для обращения к которому используется свойство



Items (список строк) относится к классу TStrings и ему присущи все свойства и методы данного класса. Свойство Items имеет кнопку [...] для открытия окна списка строк (рис.4.3).

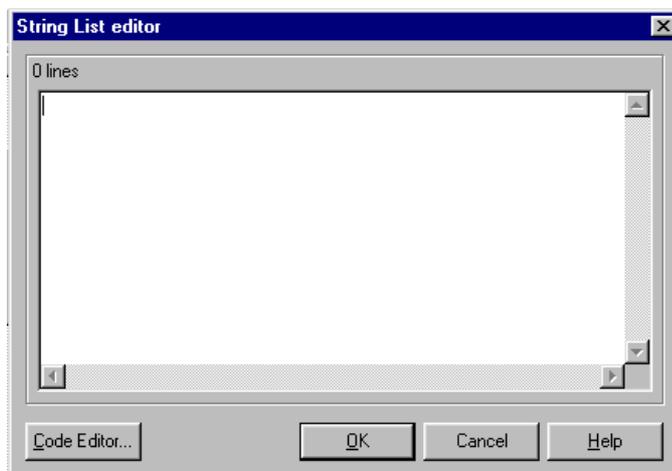


Рис. 4.3. Окно списка строк компонента TListBox.

В окне списка набираются строки, каждая из которых **индексируется, начиная с нуля.**

В нашей программе у компонента ListBox1 свойство *Items* равно:

```
«Эксперт», «КП»;  
«Власть», «КП»;  
«КП»;  
«КП», «АиФ», «АМК», «Коммерсант»;  
«КП», «МК», «Деньги», «Тема»;  
«КП – общий выпуск», «Персональный компьютер»;  
нет изданий
```

Для выделения строки в списке служит свойство *itemindex*:

- = -1 при отсутствии выделения;
- = 0, 1, 2, ... - номер выделенной строки.

Первоначально окно строк невидимо, поэтому `visible = false`.

## Задание 2.

2.1. В Delphi на форме расположите указанные выше компоненты.

2.2. Для ListBox1 выделите первую строку (какой номер она будет иметь?).

2.3. Сохраните модуль и проект.

2.4. Запишите в тетради свойства TListBox: *Items*, *ItemIndex*.

Этап 3. Алгоритмизация программирование

Оператор выбора не является классической алгоритмической структурой, поэтому не имеет установленной блок-схемы. При реализации алгоритма будем использовать словесной способ его представления.

*Алгоритм события "OK"*

**переменная: n: целого типа;**

**начало**

ввод данных в Edit1 (номер недели) - переменная n;

**выбор** значения переменной n из списка:

1 (понедельник): в ListBox1 выделить строку [0];

2 (вторник): в ListBox1 выделить строку [1];

3 (среда): в ListBox1 выделить строку [2];

4 (четверг): в ListBox1 выделить строку [3];

5 (пятница): в ListBox1 выделить строку [4];

6 (суббота): в ListBox1 выделить строку [5];

7 (воскресенье): в ListBox1 выделить строку [6];

**иначе** снять выделение (неверный день недели);

**конец выбора;**

сделать видимым ListBox;

```
Procedure TF1.Button1Click ();
```

```
var n: integer;
```

```
begin
```

```
n:=StrToInt(edit1.text);
```

```
case n of
```

```
1:ListBox1.itemindex:=0;
```

```
2:ListBox1.itemindex:=1;
```

```
3:ListBox1.itemindex:=2;
```

```
4:ListBox1.itemindex:=3;
```

```
5:ListBox1.itemindex:=4;
```

```
6:ListBox1.itemindex:=5;
```

```
7:ListBox1.itemindex:=6;
```

```
else ListBox1.itemindex:= -1;
```

```
end;
```

```
ListBox1.visible:=true;
```

<b>конец</b>	<b>end;</b>
<b>Дополнительное событие</b>	
<i>Алгоритм события "Очистить"</i>	<b>Procedure</b> TF1. Button2Click(...);
<b>начало</b>	<b>begin</b>
Очистить Edit1;	Edit1.clear;
Поместить курсор в Edit1;	Edit1.setfocus;
Снять выделение строк в ListBox1;	ListBox1.itemindex:=-1;
Сделать невидимым ListBox1;	ListBox1.visible:=false;
<b>Конец</b>	<b>end;</b>

**Задание 3.** В тетради запишите алгоритм и программу события «ОК». Создайте событие «ОК». По-возможности, создайте событие «Очистить».

Этап 4. Тестирование и отладка

После запуска программы на исполнение программа проверяется на наличие структурных ошибок. При вводе дня недели понедельник (1) выделяется первая строка, воскресенье (7) - седьмая строка списка изданий и т.д. При вводе неверного дня недели (8 и др.) - выделение снимается.

**Особенность работы TListBox** изучается при работе с выполняемой программой. TListBox – набор строк, поэтому щелчок мыши на строке приводит к ее выделению. Строки нельзя редактировать и удалять.

**Задание 4.** Завершить построение исполняемой программы. Изучите особенности компонента TListBox.

### Задача 2. Программа вывода дня выхода выбранного издания.

Компоненты TRadioGroup, TMemo.

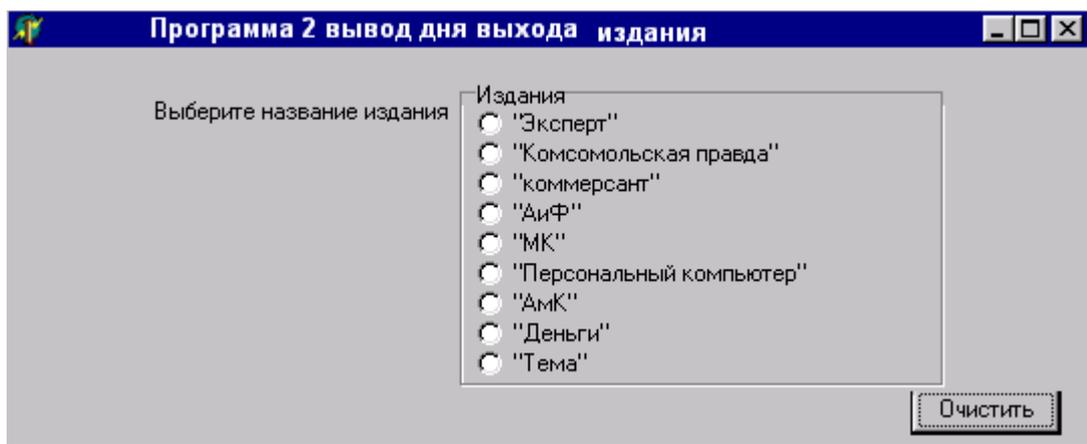


Рис.4.4. Пример окна программы 2 «Периодические издания».

## Этапы выполнения задачи 2

### Этап 1. Определение требований программы

Входные данные выбираются в TRadioGroup. Переменная *n*: *целого типа* содержит номер выделенной радиокнопки.

Выходные данные отражаются в TMemo. Переменная *d:строка* (*d:String*) определяет день выхода издания.

Дни выхода изданий:

- |                                     |                                     |
|-------------------------------------|-------------------------------------|
| ✓ «Эксперт» - понедельник, четверг; | ✓ «Персональный компьютер» - среда; |
| ✓ «КП» - вторник, пятница;          | ✓ «АМК» - понедельник;              |
| ✓ «Коммерсант» - суббота;           | ✓ «Деньги» - пятница;               |
| ✓ «АиФ» - понедельник;              | ✓ «Тема» - вторник.                 |
| ✓ «МК» - среда, пятница;            |                                     |

**Задание 1.** В тетради опишите переменные задачи 2 и их тип.

### Этап 2. Окно будущей программы

Для решения задачи расположим компоненты на окне формы (рис. 4.5).

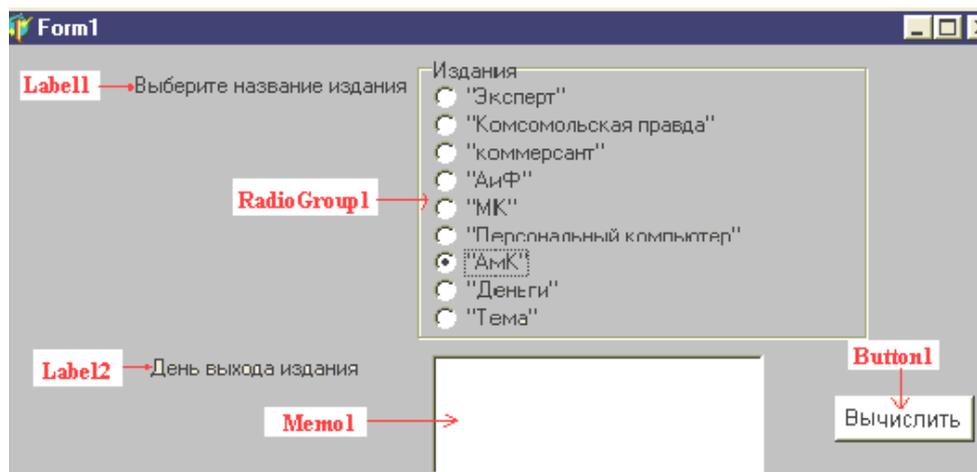


Рис. 4.5. Пример визуального проектирования окна программы 2.

В окне будущей программы располагаются компоненты TMemo, TLabel, TButton, TRadioGroup.

Форма: name = F2, Caption = Программа 2 вывод дня выхода издания

Метки: name = Label1, Caption = Выберите название издания

Name = Label2, Caption = День выхода издания

Командная кнопка: name = Button1, Caption = Вычислить

Список радиокнопок TRadioGroup аналогичен TListBox. Список радиокнопок задается свойством **Items:Tstrings**, каждая радиокнопка индексируется, начиная с нуля. Свойство Items содержит строки:

Многострочное окно ввода-вывода ТМемо представляет небольшой текстовый редактор. В окне ТМемо можно выводит, вводить, редактировать текст. В нашей программе Мемо1 служит для вывода результата.

Для обращения к строкам служит свойство Lines:Tstrings, при обращении к которому открывается окно строк. Окно вывода первоначально невидимо, т.е. *Visible=false*.

Свойства RadioGroup1.Items и Мемо1. Lines относятся к классу TStrings.

**Задание 2.** Сформируйте окно формы. В тетради запишите назначение TRadioGroup (свойства ItemIndex, Items) и ТМемо (свойства Lines).

- «Эксперт»
- «КП»
- «Коммерсант»
- «АиФ»
- «МК»
- «Персональный компьютер»
- «АмК»
- «Деньги»
- «Тема»

### Этап 3. Алгоритмизация и программирование

<p><i>Алгоритм события «Вычислить»</i></p> <p><b>переменные</b> n: целого типа; d: строка;</p> <p><b>начало</b> ввод данных n = номер выделенной радиокнопки;</p> <p><b>выбор</b> значения переменной n из списка:          [0]: d:= 'понедельник, четверг';          [1]: d:= 'вторник, пятница';          [2]: d:= 'суббота';          [3]: d:= 'понедельник';          [4]: d:= 'среда, пятница';          [5]: d:= 'среда';          [6]: d:= 'понедельник';          [7]: d:= 'пятница';          [8]: d:= 'вторник';</p> <p><b>конец выбора;</b> сделать видимым Мемо1;</p> <p><b>конец</b></p>	<pre> <b>Procedure</b> TF2. Button1Click (...); <b>var</b> n: integer; d:string; <b>begin</b> n:=RadioGroup1.itemindex; <b>case</b> n of 0: d:= 'понедельник, четверг'; 1: d:= 'вторник, пятница'; 2: d:= 'суббота'; 3: d:= 'понедельник'; 4: d:= 'среда, пятница'; 5: d:= 'среда'; 6: d:= 'понедельник'; 7: d:= 'пятница'; 8: d:= 'вторник'; <b>end;</b> Memo1.visible:=true; <b>end;</b> </pre>
<p><b>Дополнительное событие</b></p> <p><i>Алгоритм для события "Очистить"</i></p> <p><b>начало</b> снять выделение радиокнопок в RadioGroup1; очистить Мемо1; сделать невидимым Мемо1;</p> <p><b>конец</b></p>	<pre> <b>procedure</b> TF2.Button1Click(...); <b>begin</b> RadioGroup1.itemindex:=-1; Memo1.Lines.clear; Memo1.visible:=false; <b>end;</b> </pre>

**Задание 3.** В тетради запишите алгоритм и программу события "Вычислить". Создайте данное событие и реакцию на него. По-возможности создайте событие «Очистить».

#### Этап 4. Тестирование и отладка

Правильно написанная программа должна использовать для ввода – Radiogroup1, для вывода – Memo1.

Например, при выборе радиокнопки «КП» в окне вывода должны отражаться дни «вторник, пятница». Для издания «АиФ» результатом является «понедельник».

**Особенности работы TRadioGroup** удобно изучать после запуска программы. Если первоначально включить радиокнопку третьей строке, а далее включить радиокнопку пятой строки, то при этом третья радиокнопка автоматически выключится. *В данный момент времени может быть включена максимально одна радиокнопка.* Радиокнопка - "зависимый переключатель", состояние одной радиокнопки влияет на состояние других.

По-существу, радиокнопки и флажки выполняют одинаковые функции: выбор данных (параметров). Однако, принято использовать флажок как независимый, а радиокнопку как зависимый переключатели. Если одна радиокнопка включена, то остальные выключены. При включение другой, предыдущая автоматически отключается.

**Особенности работы TMemo:** 1) щелчок мыши внутри компонента (текстовой области) приводит к появлению курсора; 2) текст TMemo можно редактировать: выделять фрагмент, копировать (CTRL+INSERT), вставлять (SHIFT+INSERT), удалять и т.д.

#### **Задание 5.**

5.1. Запустите программу на исполнение. Изучите особенности работы TRadiogroup. Запишите в тетрадь понятие «зависимый переключатель».

5.2. В программе Word выберите Формат/Шрифт и Вид/Масштаб. Объясните, почему для задания эффектов шрифта используются флажки, а для указания масштаба - радиокнопки.

5.3. Изучите особенности TMemo и запишите в тетрадь.

5.4. Перечислите различия между TMemo и TListBox, TRadioGroup и TCheckBox, TMemo и TEdit, TListBox и TRadioGroup.

#### Упражнения

1. Составить программу, в которой **вывод данных** отображается включением радиокнопки:

*Уровень начальный*

а) Человек в течение своей жизни проходит несколько возрастных этапов (младенец, ребенок, подросток, ...). Требуется по введенному количеству лет определить возрастной промежуток для данного возраста.

б) В пятиэтажном доме на первом этаже 3 квартиры, на следующих этажах по четыре квартиры. Нумерация квартир с 1 по 19. Требуется по номеру квартиры указать номер этажа, на котором она расположена.

в) По введенному номеру месяца выделить номер квартала.

*Уровень средний*

г) По введенному количеству лет вывести возможные виды кредита:

до 1 года – экспресс-кредитование;

до 5 лет – кредит на неотложные нужды;

до 15 лет – ипотечное кредитование.

Для предлагаемого кредита в ТМето выводятся условия его получения.

*Уровень высокий*

д) Средний ежеквартальный расход электроэнергии - Р кВт/час. При этом:

в 1 квартале расход на 15% выше среднего,

во 2 и 4 кварталах - на 5% выше среднего,

в 3 квартале - на 8% ниже среднего.

Вводится номер квартала. Выделяется корректирующий сезонный процент и вычисляется расход электроэнергии в данном квартале.

2. Составить программу, в которой **ввод данных** осуществляется выделением строки списка.

*Уровень начальный*

а) Для выбранного дня недели вывести часы приема врача:

- |                            |                         |
|----------------------------|-------------------------|
| ▪ понедельник 8-00 – 10-00 | ▪ пятница 11-00 – 13-00 |
| ▪ вторник 11-00 – 13-00    | ▪ суббота 9-00 - 10-00  |
| ▪ среда 9-00 - 11-00       | ▪ воскресенье выходной  |
| ▪ четверг 8-00 – 10-00     |                         |

б) Программа – киноафиша: на экране список фильмов, при нажатии на одно из названий, появляется время сеансов.

в) Для выбранного автомобиля вывести данные мощности двигателя.

*Уровень средний*

г) Оклад служащего составляет X рублей. Для выбранного района (центральный, дальневосточный, северный) вычисляется з/плата. Выводится процентная надбавка (0%, 50%, 100%) и зарплата за месяц.

д) Напишите программу расчета стоимости телефонного разговора. Известно, что тариф днем, вечером и на выходных разный.

День – 100%

Вечер – скидка 20%

Ночь – скидка 50%.

Пользователь вводит количество минут и стоимость 1 минуты разговора, выбирает в списке строк категорию времени, получает результат - стоимость разговора. Результат вывести в TListBox.

*Уровень высокий*

е) Для суммы S в списке радиокнопок выбирается вид вклада («Пенсионный» - 17% годовых, «До востребования» - 3% годовых, «Молодежный» - 14% годовых и др.). Указать сумму вклада через один, два, три года с учетом начисляемых процентов.

ж) Составьте программу стоимости услуг парикмахерской. Пользователь в предложенном списке строк выбирает вид услуги, после нажатия «ОК» высвечивается стоимость услуги. Предусмотреть флажок «Скидка», для утренних клиентов 5%.

## Тема №6: Программы циклической структуры

### Циклические операторы

**Циклические операторы** содержат повторяющиеся команды.

Циклические операторы бывают трех видов:

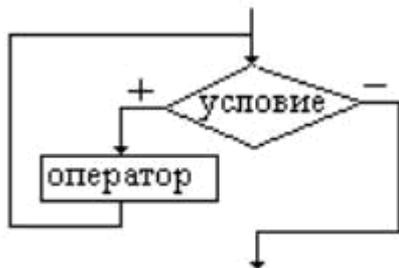
*Цикл с предусловием*

Структура оператора цикла с предусловием следующая:

**while** условие **do** оператор;

Здесь **while**, **do** – зарезервированные слова (*пока [выполняется условие], делать*).

Блок-схема цикла с предусловием следующая:



Циклический оператор с предусловием работает по следующему алгоритму. Первоначально проверяется условие. Если оно истинно, то выполняется тело цикла (оператор). Затем снова проверяется условие и т.д. Если условие ложно, то цикл завершается и выполняется оператор, стоящий непосредственно после цикла.

Если тело цикла состоит из нескольких операторов, то используется составной оператор.

*Цикл с постусловием*

Структура оператора цикла с постусловием следующая:

**repeat** тело цикла **until**;

Блок-схема цикла с постусловием следующая:



Циклическая структура цикла с постусловием работает по следующему алгоритму: сначала выполняется тело цикла (операторы), потом проверяется условие. Если условие ложно, то снова выполняются операторы. И так до тех пор, пока условие не примет значение *TRUE*.

#### *Цикл с параметром*

Структура счетного цикла (с параметром) следующая:

**for** <параметр цикла>:=<нач.знач.> **to** <кон. знач.> **do** <оператор>;

Циклическая структура счетного цикла работает по следующему алгоритму: параметру цикла присваивается начальное значение и выполняется тело цикла. Затем параметру цикла присваивается следующее значение, и снова выполняется тело цикла. И так до тех пор, пока не будут перебраны все значения параметра цикла. Каждый раз параметр цикла увеличивается на единицу.

Другая структура счетного оператора:

**for** <параметр цикла>:=<старш. знач.> **downto** <млад. знач.> **do** <оператор>;

Циклическая структура данного счетного цикла работает по следующему алгоритму: параметру цикла присваивается начальное (старшее) значение и выполняется тело цикла. Затем значение параметра уменьшается на единицу, и снова выполняется тело цикла. И так до тех пор, пока не будут перебраны все значения параметра цикла до конечного (младшего) значения.

### **Лабораторная работа №5**

#### **Вопросы теоретического контроля**

1. Понятие цикла.
2. Виды циклических операторов.
3. Цикл с предусловием: принцип работы, блок-схема, оператор.
4. Цикл с постусловием: принцип работы, блок-схема, оператор.
5. Цикл с параметром: принцип работы, блок-схема, оператор, два способа записи.
6. Два различия использования циклов с пост - и предусловиями.

#### **Цель занятия**

УЗНАТЬ:

- переменные, используемые в вычислительных циклических задачах;
- начальные значения суммы ряда;
- два различия между циклами с пост- и предусловиями;
- особенность цикла с параметром.

НАУЧИТЬСЯ:

- вычислять сумму и среднее значение числовых рядов;

**Задача:** Программа вычисления числового ряда. Стоимость производственной базы (ПБ) в 20003 г. -V тыс. руб. Ежегодный темп прироста n%. Рассчитать стоимость производственной базы на 5 лет вперед и среднее значение ряда.

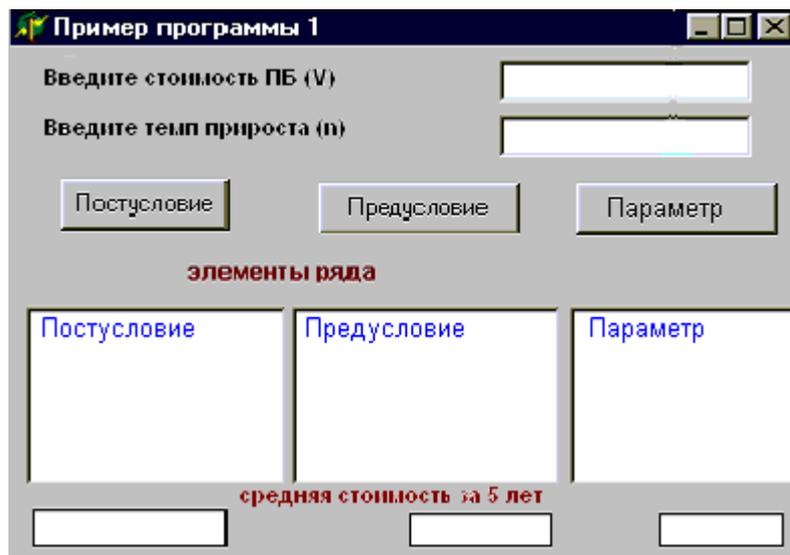


Рис. 5.1. Пример окна программы вычисления числового ряда.

### Этапы выполнения задачи

#### Этап 1. Определение требований программы

##### Входные данные

v: вещественного типа – элемент ряда (стоимость ПБ).

n: целого типа – темп прироста.

##### Промежуточные переменные

i: целого типа - параметр цикла.

##### Результат

S: вещественного типа - сумма ряда.

c: вещественного типа – среднее значение ряда.

#### ВЫЧИСЛЕНИЕ ЭЛЕМЕНТА РЯДА.

Стоимость производственной базы V в следующем году будет рассчитываться:  $V=V*(1+n/100)$ ,

где

$1+n/100$  – темп роста,

$V(\text{стоимость ПБ в новом году}) = V(\text{стоимости ПБ в прошлом году}) * \text{темпы роста}$ .

Пусть  $V_{1999} = 3$  тыс. руб. и  $n = 5\%$ . Темпы роста составят  $1 + 5/100 = 1,05$ .

$V_{2000} = V_{1999} * 1,05$ .

**Задание 1.** Опишите в тетради переменные задачи, их тип и назначение. Вспомните определение темпа роста и прироста.

### Этап 2. Окно будущей программы

Окно формы программы 1 создается согласно рис. 5.1. и 5.2.

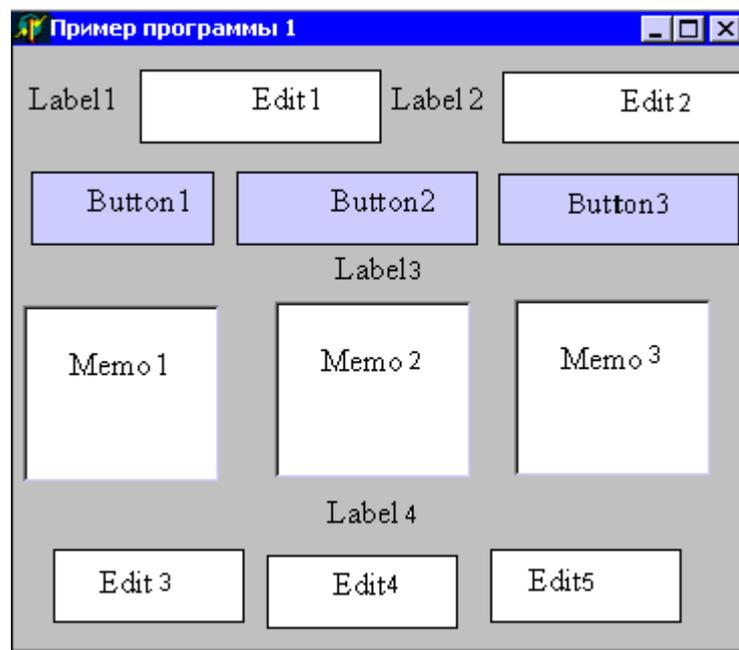


Рис. 5.2. Пример визуального проектирования окна программы 1.

**Задание 2.** В Delphi на форме расположите компоненты программы 1.

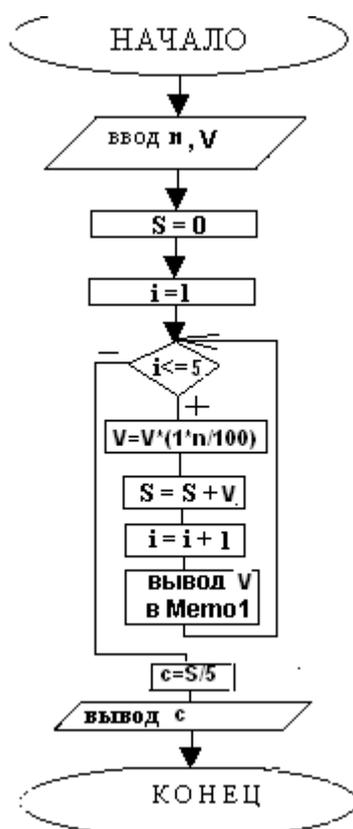
### Этап 3. Алгоритмизация и программирование

Алгоритм вычисления числового ряда содержит действия:

- 1) ввод данных;
- 2) задание первоначальных значений переменных;
- 3) тело цикла:
  - вычисление элемента ряда;
  - прибавление его к сумме;
  - изменение параметра цикла;
- 4) вывод результата

Запомните! Первоначальное значение суммы  $S=0$ , произведения  $P=1$ . Данные значения не влияют на дальнейшие вычисления.

### Алгоритм цикла с предусловием



```
Procedure TForm1.Button1Click();
```

```
var n,i:Integer;
```

```
V,c,S:real;
```

```
begin
```

```
V:=StrToInt(edit1.text);
```

```
n:=StrToInt(edit2.text);
```

```
S:=0;
```

```
i:=1;
```

```
while i<=5 do begin
```

```
V:=V*(1+n/100);
```

```
S:=S+v;
```

```
i:=i+1;
```

```
memo1.lines.add(FloatToStr(V));
```

```
end;
```

```
c:=S/5;
```

```
edit3.text:= FloatToStr(c);
```

```
end;
```

Циклы с пред- и постусловием имеют отличия:

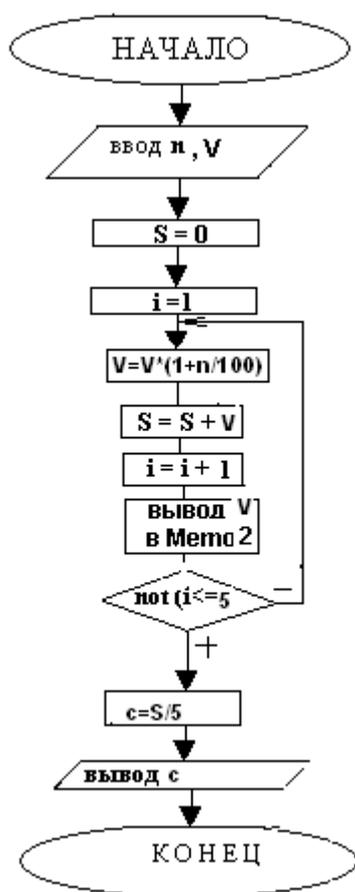
Цикл с предусловием:

- условие проверяется в начале цикла;
- тело цикла выполняется в случае истинности условия.

Цикл с постусловием:

- условие проверяется в конце цикла;
- тело цикла выполняется в случае ложности условия.

Алгоритм цикла с постусловием



```
Procedure TForm1.Button2Click();
```

```
var n,i:Integer;
```

```
V,c,S:real;
```

```
begin
```

```
V:=StrToInt (edit1.text);
```

```
n:=StrToInt (edit2.text);
```

```
S:=0 ;
```

```
i:=1;
```

```
Repeat
```

```
V:=V*(1+n/100);
```

```
S:=S+v;
```

```
i:=i+1;
```

```
memo2.lines.add(FloatToStr(v));
```

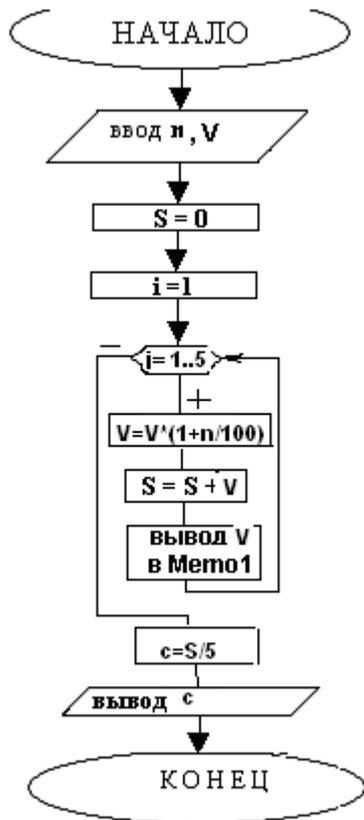
```
Until not (i<=5);
```

```
C:=S/5;
```

```
Edit4.text:= FloatToStr(c);
```

```
end;
```

Цикл с параметром не является классической алгоритмической структурой, поэтому для него нет определенной блок-схемы.



- Цикл с параметром имеет два отличия:
- параметр цикла изменяется строго на 1;
  - изменение параметра цикла происходит автоматически.

**Procedure TForm1.Button3Click();**

**var**

n,i:Integer;

V,c,S:real;

**begin**

V:=StrToInt (edit1.text);

n:=StrToInt (edit2.text);

S:=0; i:=1;

**For i:=1 to 5 do begin**

V:=V\*(1+n/100);

S:=S+V;

memo3.lines.add(FloatToStr(V));

{ i:=i+1; - отсутствует, параметр цикла изменяется автоматически}

**end;**

**For i:=5 downto 1 do begin**

V:=V\*(1+n/100);

S:=S+V;

memo3.lines.add(FloatToStr(V));

**end;**

c:=S/5;

edit5.text:= FloatToStr(c);

**end;**

**Задание 3.**

3.1.Зарисуйте блок-схемы и программы трех видов циклов в тетрадь.

3.2. Создайте события программы и реакцию на него.

3.3. Выпишите отличия циклов с пред- и постусловиями, особенность цикла с параметром. 3.4. Для цикла с параметром реализуйте один из двух способов записи.

#### Этап 4. Тестирование и отладка

Задача 1 решается тремя способами с использованием различных видов циклов. При этом результаты всех вариантов решения должны совпадать.

При тестировании задаются входные данные и рассчитывается результат.

Пусть  $V_{1999}=30$  тыс. руб.,  $n=7\%$ .

$V_{2000}=\underline{\hspace{2cm}}$

$V_{2001}=\underline{\hspace{2cm}}$

...

$V_{2004}=\underline{\hspace{2cm}}$

Сумма за 5 лет равна  $S=\underline{\hspace{2cm}}$

Средняя стоимость ПБ  $s=\underline{\hspace{2cm}}$

#### Задание 4.

4.1. Рассчитайте на бумаге результат задачи при  $V=20$ ,  $n=5\%$ .

4.2. Запустите программу на исполнение. Все алгоритмы задачи должны выводить одинаковые результаты, согласно Вашим расчетам.

#### Упражнения

##### *Уровень начальный*

1. Начальный объем выпуска предприятия составляет  $S$  ед. Ежегодное увеличение выпуска составляет  $k\%$  к предыдущему году. Рассчитать предполагаемый выпуск предприятия на  $t$  лет вперед и суммарный выпуск предприятия.
2. Объем продажи авиабилетов в 2002 г. составил  $X$  тыс. ед. Ежегодный темп прироста составляет  $n\%$ . Рассчитать объем продажи до 2005 г. и средний ежегодный объем продаж.
3. В магазине №1 спрос на хозяйственные товары в 2001 г. составил  $m$  тыс. руб. Ежегодный прирост  $n\%$ . Рассчитать спрос на 3 года вперед и средний спрос.

##### *Уровень средний*

4. Объем продажи авиабилетов в 1 квартале 2002 г. составил  $X$  тыс. ед. Ежеквартальный темп прироста составляет  $n\%$ . Рассчитать объем продажи до конца 2002 г. и средний ежеквартальный объем продаж.
5. Человек открыл счет в банке  $S$  руб. под  $n\%$  годовых. Начисление процентов происходит в конце года. Рассчитать сумму вклада на 3 года вперед и среднее значение вклада за 3 года.

6. Первоначальная стоимость основного фонда предприятия составляет  $S$  млн. руб. Амортизация в год составляет  $n\%$ : ежегодная стоимость уменьшается на  $n\%$ . Рассчитать стоимость основного фонда предприятия на  $t$  лет вперед.

*Уровень высокий*

7. Для кредита на сумму  $S$  на  $t$  лет под  $n\%$  годовых рассчитать сумму выплат: погашение основного кредита + выплата процентов. Известно, что сумма основного долга выплачивается равными долями ежемесячно, ежемесячный процент  $n/12$  рассчитывается на оставшуюся сумму долга. Указать общую сумму выплаты и сумму переплаты по отношению к начальной сумме.
8. В 1 квартале 2000 г. цена на телевизор «SAMSUNG-2536NY» составила  $x$  тыс. руб. Ежеквартальное увеличение цены составляет  $n\%$ . Вычислить ежеквартальный спрос на TV -  $y$  единиц, если  $y=3,4*x^{0,4}$ .

### **Дополнительные задания на повторение**

Дополните выполненное упражнение:

*Уровень начальный*

- а) панелью радиокнопок с выбором цвета формы;
- б) набором строк с выбором размера формы;
- в) панелью радиокнопок с выбором заглавия формы.

*Уровень средний*

- г) флажком с выбором «курсив» в Tmemo;
- д) флажком с выбором «красный цвет шрифта» в TLabel;

*Уровень высокий*

- е) набором строк с выбором «Удалить строку в мемо1», «Сохранить мемо2 в файл», «Добавить строку 'ЭЛЕМЕНТ' в мемо3», «Вывести окно диалога в ФИО автора».

### **Творческие задания**

1. Начальный объем выпуска предприятия составляет  $S$  ед. Ежегодное увеличение выпуска составляет  $k\%$  к предыдущему году:  $S_t = S_{t-1} * k$ . Рассчитать предполагаемый выпуск предприятия на  $t$  лет вперед.
2. Для кредита на сумму  $S$  на  $t$  лет под  $n\%$  годовых рассчитать сумму выплат: погашение основного кредита и выплата процентов. Известно, что сумма основного долга выплачивается равными долями ежемесячно, ежемесячный процент  $n/12$  рассчитывается на оставшуюся сумму долга.

Первоначальная стоимость основного фонда предприятия составляет S млн. руб. Амортизация в год составляет n%: ежегодная стоимость уменьшается на n%. Рассчитать стоимость основного фонда предприятия на t лет вперед.

## Тема №7: Программы работы с векторами

### Массивы

**Массивы** – упорядоченный набор однотипных элементов. Все элементы должны быть одного и того же типа.

В Object Pascal возможна работа с несколькими видами массивов:

- одномерными;
- двумерными;
- динамическими.

#### *Одномерные массивы*

В одномерном массиве компоненты представлены в виде одной строки (одного столбца).

Все компоненты одного типа. Каждый компонент имеет свой индекс.

Описание одномерного массива возможно двумя способами:

1. В разделе описания типов:

**type**

<имя типа массива> = **array** [<тип индекса>] **of** <тип компонент>;

**var**

<имя массива>: <имя типа массива>;

2. В разделе описания переменных:

**var**

<имя массива>: **array** [<тип индекса>] **of** <тип компонент>;

Обычно в качестве индексного типа используется тип-диапазон, в котором задаются границы изменения индексов. Например:

*Type*

*Tnumber = array [0..19] of integer;*

*Var number: Tnumber;*

или

**var number: array [0..20] of integer;**

К конкретному элементу массива можно обратиться по имени и индексу. При этом индекс записывается в квадратные скобки, например: number[0], number[2], ..., number[19].

### Лабораторная работа №6.

#### Вопросы теоретического контроля

1. Понятие одномерного массива (вектора).

2. Описание переменной типа вектор. Индексация элементов вектора.

3. Обращение к элементу вектора.

### Цель занятия

узнать:

- способы ввода-вывода вектора;
- задание значений элементов вектора;
- алгоритм сравнения элементов вектора.

научиться:

- суммировать элементы вектора;
- находить количество элементов с заданными свойствами;
- находить max (min) элемент вектора;
- формировать вектор-результат.

**Задача:** Программа «Курс доллара 2003 г.». Вычисление:

- среднегодового курса доллара;
- количества месяцев курса выше 30 руб.;
- максимальное значение курса доллара в году;
- вектора-результата как разница между средним и текущим значениями \$.

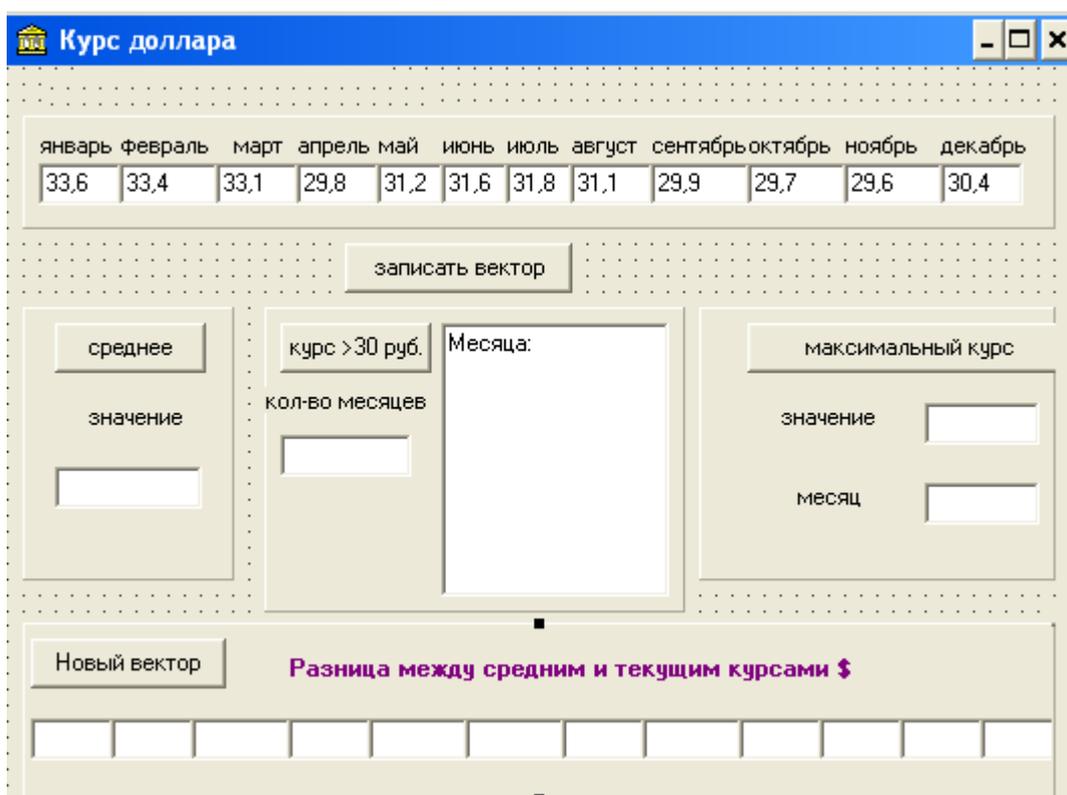


Рис.6.1. Окно программы задачи «Курс доллара 2003 г.».

### Этапы выполнения задачи 1

#### **Этап 1. Определение требований программы**

В программе используются переменные:

Входные:  $a: \text{array} [1..12] \text{ of real}$ ; - вектор ежемесячных значений курса \$ (12 месяцев).

Промежуточные: `i:integer`; - счетчик.

Выходные:

- ✓ `S, c:real`; - сумма и среднее элементов вектора;
- ✓ `k:integer`; - количество месяцев с курсом \$ более 30 руб.;
- ✓ `max:real`; - максимальный курс \$;
- ✓ `nmax:integer`; - номер месяца максимального курса.
- ✓ `b:array [1..12] of real`; - вектор-результат.

**Задание 1.** В тетради запишите условие задачи и переменные.

### Этап 2. Окно будущей программы

Для решения задачи форма формируется согласно рис.6.1 и 6.2. Вывод результатов каждой подзадачи осуществляется на отдельную панель.

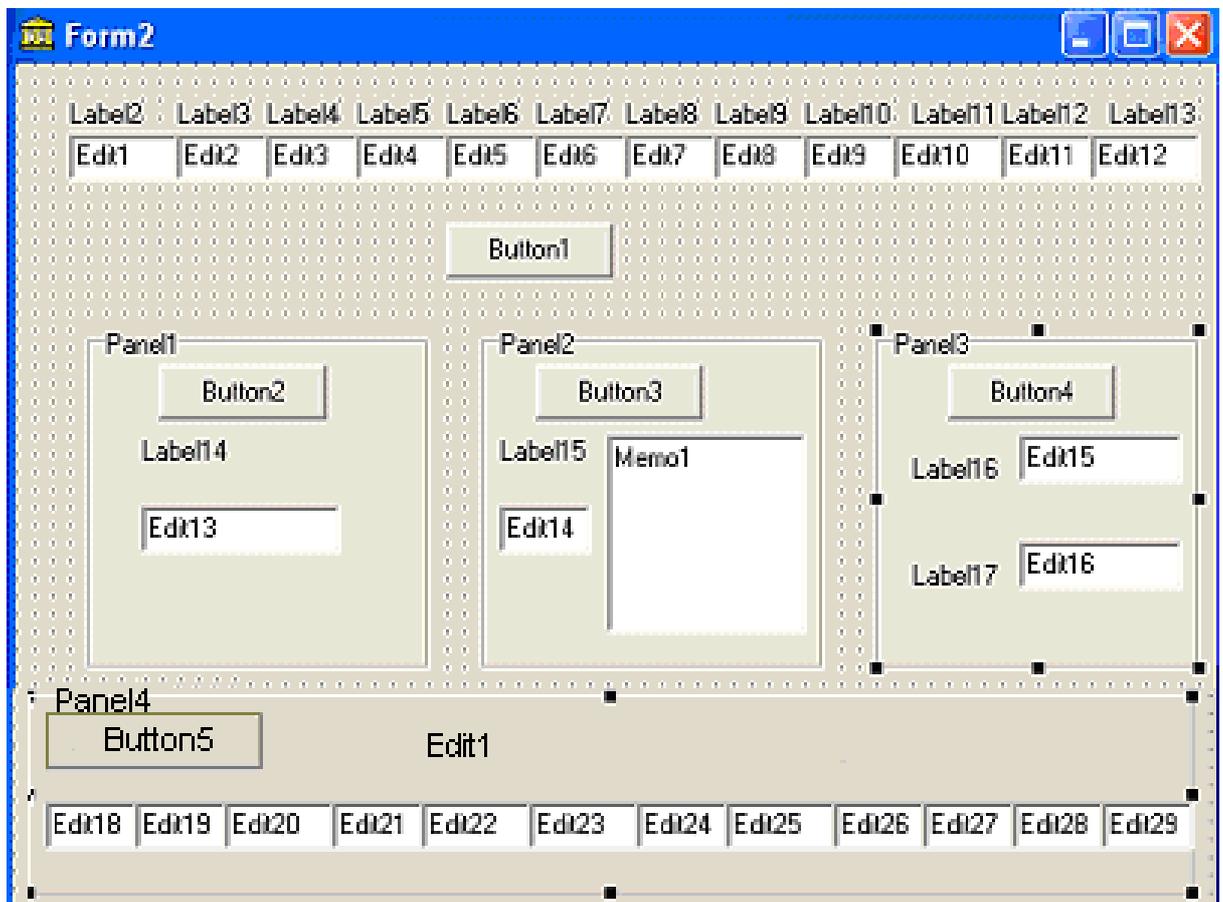


Рис.6.2. Компоненты окна формы задачи «Курс доллара 2003 г.».

**Задание 2.** В Delphi создайте окно будущей программы (окно формы).

### Этап 3. Алгоритмизация и программирование

#### *Локальное и глобальное описание переменных.*

В ранее рассматриваемых задачах переменные описывались внутри каждой процедуры – локальное описание. *Локальные переменные известны только внутри данной процедуры.*

Как уже рассматривалось в лаб. работе №2 , глобальное описание данных осуществляется в первой части модуля - INTERFACE. Глобальные переменные действительны для всех процедур модуля.

В настоящей задаче исходный вектор является глобальным данным. Он будет использоваться во всех процедурах модуля. Переменная среднегодового значения курса \$ также будет глобальной – будет использоваться в двух процедурах. Мы можем все переменные описать как глобальные, при этом оперативная память, используемая работающей программой, увеличится на количество байт, необходимых для хранения значений глобально описанных переменных. В целях экономии оперативной памяти при написании программных продуктов количество данных (в т.ч. переменных) минимизируется.

**Unit ...**

**Interface**

...

**procedure** TForm1.Button1Click(Sender: TObject); («Записать вектор»)

**procedure** TForm1.Button2Click(Sender: TObject); («Средний курс \$»)

**procedure** TForm1.Button3Click(Sender: TObject); («Количество месяцев с курсом более 30 руб.»)

**procedure** TForm1.Button4Click(Sender: TObject); («Максимальный курс»)

**procedure** TForm1.Button5Click(Sender: TObject); («Новый вектор»)

**var** Form1: TForm1;

a:array [1..12] of real;

S,c: real;

i:integer; k:integer;

max:real; nmax:integer;

b:array [1..12] of real;

Переменные описаны в секции INTERFACE

Все переменные являются глобальными

**Implementation**

**procedure** TForm1.Button1Click(Sender: TObject);

**begin**

a[1]:=StrToFloat(Edit2.text);

a[2]:=strToFloat(edit3.text);

a[3]:=strToFloat(edit4.text);

a[4]:=strToFloat(edit5.text);

a[5]:=strToFloat(edit6.text);

a[6]:=strToFloat(edit7.text);

a[7]:=strToFloat(edit8.text);

**«Записать вектор»**

**начало**

Элементам вектора присвоить значения с компонентов TEdit.

<pre>a[8]:=strToFloat(edit9.text); a[9]:=strToFloat(edit10.text); a[10]:=strToFloat(edit11.text); a[11]:=strToFloat(edit12.text); a[12]:=strToFloat(edit13.text); end;</pre>	<p><b>конец;</b></p>
<pre>procedure TForm1.Button2Click(Sender: TObject); begin S:=0; for i:=1 to 12 do     S:=S+a[i]; c:=S/12; edit13.text:=FloatToStr(c); end;</pre>	<p><b>«Средний курс \$»</b></p> <p><b>начало</b> первоначально сумма \$ S:=0; Цикл: перебирая все месяца к сумме прибавляем текущий курс \$; Среднее c:=S/12; В edit 13 вывести результат c - среднее;</p> <p><b>Конец</b></p>
<pre>procedure TForm1.Button3Click(Sender: TObject); begin k:=0; for i:=1 to 12 do     if a[i]&gt;30 then begin         k:=k+1;         memo1.Lines.Add(IntToStr(i));     end; edit14.Text:=IntToStr(k); end;</pre>	<p><b>«Количество месяцев с курсом более 30 руб.»</b></p> <p><b>начало</b> Первоначально кол-во найденных месяцев k=0; <b>Цикл:</b> перебирая все месяца, <b>если</b> курс i-месяца a[i]&gt;30, <b>то начало</b> увеличиваем кол-во найденных месяцев k:=k+1; выводим номер месяца в Memo1;</p> <p><b>конец;</b></p> <p>в Edit 14 выводится результат k;</p> <p><b>конец</b></p>

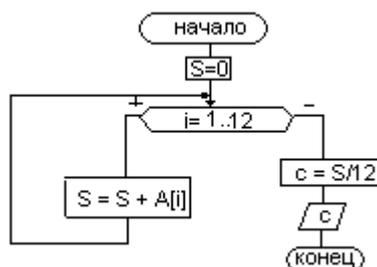
<pre>procedure TForm1.Button4Click(Sender: TObject); begin max:=a[1]; nmax:=1; for i:=1 to 12 do     if a[i]&gt;max then         begin             max:=a[i]; nmax:=i;         end; edit15.Text:=FloatToStr(max); edit16.Text:=IntToStr(nmax); end;</pre>	<p><b>«Максимальный курс»</b></p> <p><b>начало</b> Первоначально считается: max курс \$ в 1 месяце, номер месяца nmax =1. <b>Цикл:</b> перебирая все месяца, <b>если</b> курс \$ i-месяца a[i] &gt; max, <b>то</b></p> <p><b>начало</b> значения max, nmax переписываются;</p> <p><b>конец;</b></p> <p>в Edit 15 выводится max курс \$ в Edit 16 выводится номер месяца</p> <p><b>конец;</b></p>
<pre>procedure TForm1.Button5Click(Sender: TObject);</pre>	<p><b>«Новый вектор»</b></p>

<code>var B: array [1..12] of real;</code>	переменная B – новый вектор
<code>begin</code>	<b>начало</b>
<code>for i:=1 to 12 do B[i]:=c-a[[i];</code>	Цикл: перебирая все месяца, $B[i-\text{месяца}] = \text{среднее}C - A[i-\text{месяца}]$
<code>edit17.Text:=FloatToStr(B[1]);</code>	в Edit 17 выводится B[1]
<code>...</code>	...
<code>edit29.Text:=FloatToStr(B[12]);</code>	в Edit 29 выводится B[12]
<code>end;</code>	<b>конец;</b>

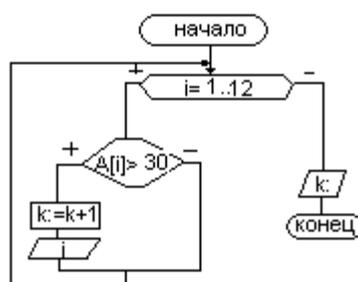
### Блок-схемы подзадач

Графическое представление линейных, циклических и разветвляющих алгоритмов Вам уже известны. Именно эти структуры используются для решения любых задач, в том числе работы с векторами. Рассмотрим две наиболее часто используемые блок-схемы решения подзадач работы с векторами.

«Средний курс \$» – блок-схема содержит циклический алгоритм.



«Количество месяцев с курсом более 30 руб.» – блок-схема содержит циклических алгоритм с вложенным неполным ветвлением.



### Задание 3.

- 3.1. Создайте события и реакции на события программы.
- 3.2. Зарисуйте блок-схемы подзадач «Запомнить», «Максимальный» и «Новый вектор».

### Этап 4. Тестирование и отладка

Данные курса \$ известны и представлены на рис. 6.1. Для проверки структуры программы рассчитывают:

- ✓ S – сумму всех элементов;

- ✓ С – среднее значение;
- ✓ К – количество элементов более 30 руб. и номера этих месяцев;
- ✓ Мах – максимальный курс \$ и номер месяца максимального курса \$.

**Задание 4.** Рассчитайте на бумаге данные результата. Сравните с результатом программы.

### Упражнения

1. Для вектора объема продаж квартир А (2000-2004 гг.) найти:

- среднегодовой объем продаж квартир;
- максимальный объем продаж и номер года максимального объема продаж;
- количество лет, в которых объем продаж квартир был ниже среднего;
- получить новый вектор, как разница между максимальным объемом и текущим годовым.

2. Для вектора расходов на оплату услуг мобильной связи В (2001-2005 гг.) найти:

- суммарный расход за 2001 – 2005 гг.;
- минимальный расход и номер года минимального расхода;
- количество лет, в которых расход был выше минимального;
- получить новый вектор, как процент текущего года от суммы.

3. Для вектора доходов от реализации Интернет-услуг С (2001-2005 гг.) найти:

- суммарный доход за 2001 – 2005 гг.;
- максимальный и минимальный доход;
- количество лет с доходом выше  $\frac{1}{4}$  суммы доходов за 2001-2005 гг.;
- получить новый вектор, в котором значения лет с доходом ниже  $\frac{1}{4}$  суммы доходов за 2001-2005 гг. заменить нулями.

### Творческие задания

*уровень начальный*

1. В векторе объема удоя молока фермерского хозяйства А (2000-2004 гг.) найти максимальное значение. Указать количество максимальных элементов и года их встречи.
2. В векторе объема продаж мороженого С (2000-2004 гг.) найти среднегодовое значение. Получить новый вектор как разницу между средним и текущим годом.

*уровень средний*

3. Для вектора выпуска бытовой техники А (2000-2004 гг.) найти года выпуска, значения которых больше среднего.
4. В векторе спроса на ЭКСПРЕСС-кредит С (2000-2004 гг.) найти суммарное и максимальное значения. Указать процент максимального значения от суммарного.

*уровень высокий*

5. Для вектора объема сбора урожая В (1995-2004гг.) найти средние значения XX(1995-2000 гг.) и XXI(2001-2004 гг.) и указать в каком веке наибольшее.
6. Для вектора В (1999-2004гг.) найти разницу между минимальным и максимальным значениями. Указать год с минимальным значением.
7. Для вектора объема продаж авиабилетов А (2000-2002 гг.) получить новый вектор как разницу между объемом 2000 г. и текущим годовым значением.

## Тема №8: Программы работы с матрицами

### *Двумерные массивы*

Двумерные массивы представляют собой **матрицу**. Элементы в матрице располагаются по строкам и столбцам.

Описание двумерного массива:

**type**

<имя типа>= **array** [<тип инд. строк, тип инд. столбцов>] **of** <тип компонент>;

**var**

<имя массива>: <имя типа массива>;

*или*

**var**

<имя массива>:**array** [<тип инд. строк, тип инд. столбцов>] **of** <тип компонент>;

Чтобы обратиться к конкретному элементу матрицы, надо указать в квадратных скобках, через запятую индекс строки, индекс столбца. В Object Pascal можно одним оператором присваивания передать все элементы одного массива другому. Например:

**Var**

A,B: **array** [1 .. 5, 1..4] **of** real;

**Begin**

A[1,1]:=1;

A[1,2]:=2;

A[1,3]:=4;

B:=A;

**End;**

## Лабораторная работа №7.

### Вопросы теоретического контроля

4. Понятие двумерного массива (матрицы).
5. Описание переменной типа матрица. Индексация элементов матрицы.
6. Обращение к элементу матрицы.

### Цель занятия

#### **узнать:**

- способы ввода-вывода матрицы;
- задание значений элементов матрицы;
- алгоритм сравнения элементов матрицы.

#### **научиться:**

- суммировать элементы матрицы;
- находить количество элементов с заданными свойствами;
- находить max (min) элемент матрицы;
- формировать матрицу-результат.

**Задача:** Программа «Поквартальный курс доллара 2003-2005 гг.». Вычисление:

- среднеквартального курса доллара;
- количества кварталов каждого года выше 30 руб.;
- максимальное значение курса доллара;
- матрицы-результата с заменой курса ниже среднеквартального на 0.

The screenshot shows a software window with a blue title bar. The main area is divided into several sections:

- Top Left:** A table with columns for years (2003, 2004, 2005) and rows for quarters (1, 2, 3, 4). Values are: 1Q: 28,3; 33,6; 33,4; 2Q: 33,1; 29,8; 31,2; 3Q: 31,6; 31,8; 31,1; 4Q: 29,9; 29,7; 29,6. A button 'запомнить матрицу' is below.
- Top Right:** A section titled 'Значения ниже среднего заменены нулями' with a similar table. The value for 2004 Q2 is 0. A button 'Новая матрица' is to the right.
- Bottom Left:** A section for 'среднеквартальное значение' with a text box containing '28,60833333'.
- Bottom Middle:** A section for 'курс >30 руб.' with three input boxes for the number of quarters in 2003 (2), 2004 (2), and 2005 (3).
- Bottom Right:** A section for 'максимальный курс' with input boxes for 'значение' (33,6), 'год' (2004), and 'квартал' (1).

Рис.7.1. Окно программы задачи «Курс доллара 2003-2005 гг.».

### Этапы выполнения задачи 1

#### **Этап 1. Определение требований программы**

В программе используются переменные:

Входные: **a:array** [1..4, 2003..2005] of real; - матрица 1-4 строки, 2003-2005 столбцы.

Промежуточные: **i:integer**; - счетчик кварталов;

**g:integer**; - счетчик лет.

Выходные:

- ✓ **S, s:real**; - сумма и среднее элементов матрицы;
- ✓ **к2003, к2004, к2005:integer**; - количество кварталов года с курсом \$ более 30 руб.;
- ✓ **max:real**; - максимальный курс \$ матрицы;
- ✓ **kvmax:integer**; - номер квартала максимального курса.
- ✓ **godmax:integer**; - номер года максимального курса.
- ✓ **b:array** [1..4, 2003.. 2005] of real; - матрица-результат.

**Задание 1.** В тетради запишите условие задачи и переменные. Укажите какие переменные глобальные, какие локальные для подзадач «Запомнить», «Среднеквартальное», «Количество», «Максимальное», «Новая матрица».

### Этап 2. Окно будущей программы

Для решения задачи форма формируется согласно рис.7.1 и 7.2. Вывод результатов каждой подзадачи осуществляется на отдельную панель.



Рис.7.2. Компоненты окна формы задачи «Курс доллара 2003 г.».

**Задание 2.** В Delphi создайте окно будущей программы (окно формы).

### Этап 3. Алгоритмизация и программирование

Unit ...

Interface (Секция описания данных)

...

**procedure** TForm1.Button1Click(Sender: TObject); («Запомнить матрицу»)

**procedure** TForm1.Button2Click(Sender: TObject); («Среднеквартальный курс матрицы»)

**procedure** TForm1.Button3Click(Sender: TObject); («Количество кварталов года с 1\$ > 30 руб.»)

**procedure** TForm1.Button4Click(Sender: TObject); («Максимальный курс»)

**procedure** TForm1.Button5Click(Sender: TObject); («Новая матрица»)

**var** Form1: TForm1;

**a:array** [1..4, 2003..2005] of real;

**i,g:integer;**

Переменные секции INTERFACE глобальные

Матрица A используется во всех процедурах

Счетчики кварталов I и лет G часто используются

«Среднеквартальное» и «Новая матрица»

**c:real;**

### Implementation (Секция реализации)

<pre><b>procedure</b> TForm1.Button1Click(Sender: TObject); <b>begin</b> a[1,2003]:=StrToFloat(Edit1.text); a[1,2004]:=strToFloat(edit2.text); a[1,2005]:=strToFloat(edit3.text); a[2,2003]:=strToFloat(edit4.text); a[2,2003]:=strToFloat(edit4.text); a[2,2005]:=strToFloat(edit6.text); a[3,2003]:=strToFloat(edit7.text); a[3,2004]:=strToFloat(edit8.text); a[3,2005]:=strToFloat(edit9.text); a[4,2003]:=strToFloat(edit10.text); a[4,2004]:=strToFloat(edit11.text); a[4,2005]:=strToFloat(edit12.text); <b>end;</b></pre>	<p><b>«Запомнить матрицу»</b></p> <p><b>начало</b></p> <p>Элементам матрицы присвоить значения с компонентов TEdit.</p> <p><b>конец;</b></p>
<pre><b>procedure</b> TForm1.Button2Click(Sender: TObject); <b>var</b> S:real; <b>begin</b> S:=0; <b>for</b> i:=1 <b>to</b> 4 <b>do</b>   <b>for</b> g:=2003 <b>to</b> 2005 <b>do</b>     S:=S+a[i,g]; c:=S/12; edit13.text:=FloatToStr(c); <b>end;</b></pre>	<p><b>«Среднеквартальный курс \$»</b></p> <p>локальная переменная S - сумма</p> <p><b>начало</b></p> <p>первоначально сумма \$ S:=0;</p> <p><b>Цикл</b> I=1..4: перебирая все кварталы</p> <p> <b>Цикл</b> G=2003..2005: перебирая все года</p> <p>    к сумме прибавляем текущий курс \$;</p> <p>Среднее c:=S/12;</p> <p>В edit 13 вывести результат c - среднее;</p> <p><b>Конец</b></p>
<pre><b>procedure</b> TForm1.Button3Click(Sender: TObject); <b>var</b> k2003, k2004,k2005:integer; <b>begin</b> k2003:=0; <b>for</b> i:=1 <b>to</b> 4 <b>do</b></pre>	<p><b>«Количество месяцев года с 1\$ &gt; 30 руб.»</b></p> <p>Локальные переменные количества каждого года</p> <p><b>начало</b> Первоначально кол-во найденных кварталов 2003 года k2003=0;</p> <p><b>Цикл</b> I=1..4: перебирая все кварталы</p>

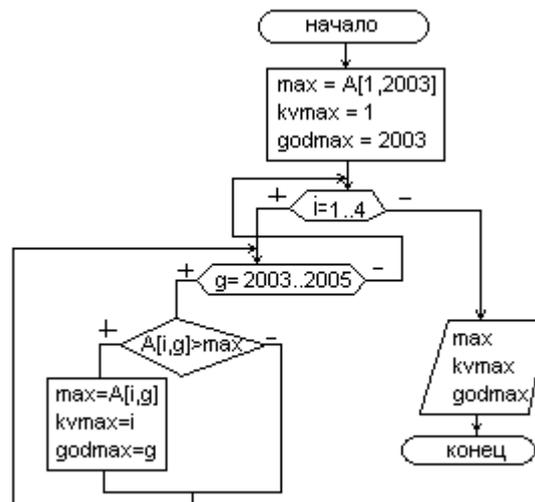
<pre> <b>for</b> g:=2003 <b>to</b> 2003 <b>do</b>   <b>if</b> a[i,g]&gt;30 <b>then</b> k2003:=k2003+1;  edit30.Text:=IntToStr(k2003);  k2004:=0; <b>for</b> i:=1 <b>to</b> 4 <b>do</b>   <b>for</b> g:=2004 <b>to</b> 2004 <b>do</b>     <b>if</b> a[i,g]&gt;30 <b>then</b> k2004:=k2004+1;   edit31.Text:=IntToStr(k2004);  k2005:=0; <b>for</b> i:=1 <b>to</b> 4 <b>do</b>   <b>for</b> g:=2005 <b>to</b> 2005 <b>do</b>     <b>if</b> a[i,g]&gt;30 <b>then</b> k2005:=k2005+1;   edit32.Text:=IntToStr(k2005); <b>end</b>; </pre>	<p><b>Цикл</b> G=2003..2005: перебирая все года</p> <p><b>если</b> курс i-квартала, g-года a[i,g]&gt;30,</p> <p><b>то</b> увеличиваем кол-во k2003:=k2003+1;</p> <p>в Edit 30 выводится результат k2003;</p> <p>...</p> <p><b>конец</b></p> <p>Эту же процедуру можно было реализовать другими способами. Например,</p> <p><b>Begin</b> k2003:=0; k2004:=0; k2005:=0;</p> <p><b>for</b> i:=1 <b>to</b> 4 <b>do begin</b></p> <p><b>if</b> a[i,2003]&gt;30 <b>then</b> k2003:=k2003+1;</p> <p><b>if</b> a[i,2004]&gt;30 <b>then</b> k2004:=k2004+1;</p> <p><b>if</b> a[i,2005]&gt;30 <b>then</b> k2005:=k2005+1;</p> <p><b>end</b>;</p> <p>edit30.Text:=IntToStr(k2003);</p> <p>edit31.Text:=IntToStr(k2004);</p> <p>edit32.Text:=IntToStr(k2005);</p> <p><b>end</b>;</p>
--	--

<pre> <b>procedure</b> TForm1.Button4Click(Sender: TObject); <b>var</b> max:real; kvmax, Godmax:integer; <b>begin</b>   max:=a[1,2003];   kvmax:=1;   Godmax:=2003;   <b>for</b> i:=1 <b>to</b> 4 <b>do</b>     <b>for</b> g:=2003 <b>to</b> 2005 <b>do</b>       <b>if</b> a[i,g]&gt;max         <b>Then begin</b>           max:=a[i,g];           kvmax:=i;           godmax:=g;         <b>end</b>;   edit14.Text:=FloatToStr(max);   edit15.Text:=FloatToStr(godmax); </pre>	<p><b>«Максимальный курс»</b></p> <p>Переменные max-курс, max-квартал, max-год</p> <p><b>начало</b> Первоначально считается:</p> <p>max курс \$ в 1 квартале 2003 года,</p> <p>max-квартал lmax =1.</p> <p>max-год Godmax =2003.</p> <p><b>Цикл</b> l=1..4: перебирая все кварталы</p> <p><b>Цикл</b> G=2003..2005: перебирая все года</p> <p><b>если</b> курс \$ i-месяца g-года a[i,g] &gt; max,</p> <p><b>то начало</b></p> <p>значения max, kvmax, godmax переписываются;</p> <p><b>конец</b>;</p> <p>в Edit 14 выводится max-курс \$</p> <p>в Edit 15 выводится max-квартал</p>
---	--

edit16.Text:=IntToStr(kvmax); <b>end;</b>	в Edit 16 выводится max-год <b>конец;</b>
<b>procedure</b> TForm1.Button5Click(Sender: TObject); <b>var</b> B: array[1..4,2003..2005] of real; <b>begin</b> <b>for</b> i:=1 to 4 <b>do</b> <b>for</b> g:=2003 to 2005 <b>do</b> <b>if</b> a[i,g]<c <b>then</b> b[i,g]:=0 <b>else</b> b[i,g]:=a[i,g];  edit18.text:=FloatToStr(b[1,2003]); edit19.text:=FloatToStr(b[1,2004]); ... edit29.text:=FloatToStr(b[4,2005]); <b>end;</b>	<b>«Новая матрица»</b> Локальная переменная B – новая матрица <b>начало</b> <b>Цикл</b> I=1..4: перебирая все кварталы <b>Цикл</b> G=2003..2005: перебирая все года <b>если</b> курс i-месяца g-года < среднего (с), <b>то</b> в новой матрице на этом месте пишем 0 <b>иначе</b> в новой матрице значение не меняется в Edit 18 выводится B[1,2003] в Edit 19 выводится B[1,2004] ... в Edit 29 выводится B[4,2005] <b>конец;</b>

### Блок-схемы подзадач

Блок-схема «Максимальный курс \$» содержит 3 структуры: в цикл для I=1..4 вложен цикл для G=2003 для 2005, который в свою очередь содержит ветвление.



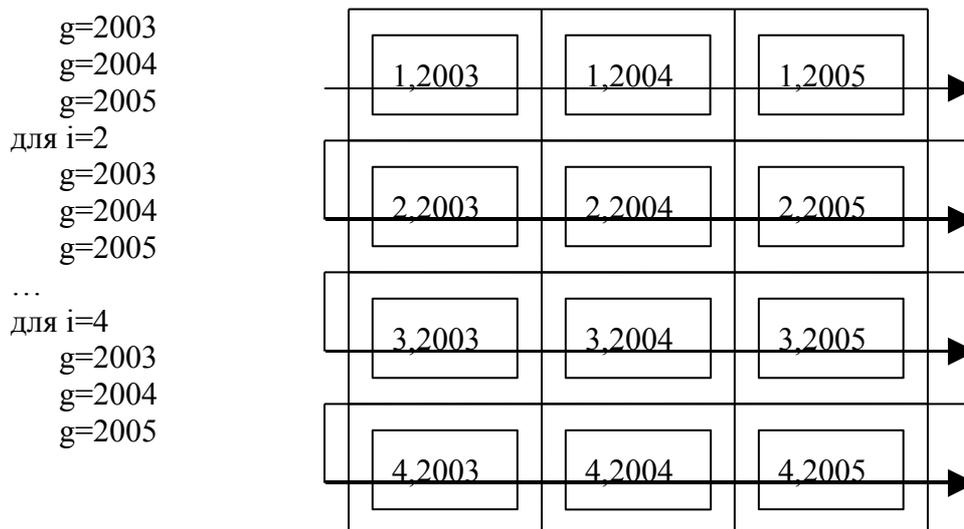
В результате выполнения данного алгоритма:

For i:= 1 to 4 do

For g:= 2003 to 2005 do ...

просмотр матрицы будет осуществляться следующим образом:

для i=1



### Задание 3.

3.1. Создайте события и реакции на события программы.

3.2. Зарисуйте блок-схемы «Количество» и «Среднеквартальное».

#### Упражнения

1. Для матрицы объема продаж квартир А (1-4 кв., 2002-2004 гг.) найти:

- средней объем продаж квартир каждого года;
- максимальный объем продаж всей матрицы, номер квартала и года максимального объема продаж;
- количество кварталов всей матрицы, в которых объем продаж квартир был ниже среднего;
- получить новую матрицу, как разница между максимальным объемом и текущим кварталом.

2. Для матрицы расходов на оплату услуг мобильной связи В (1-2 полугодия, 2001-2005 гг.) найти:

- суммарный расход за 2001 – 2005 гг.;
- минимальный расход каждого года, номер полугодия;
- количество полугодий матрицы, в которых расход был выше минимального;
- получить новую матрицу, как процент полугодия от суммы.

3. Для матрицы доходов от реализации Интернет-услуг С (1-4 кв., 2003-2005 гг.) найти:

- суммарный доход всей матрицы за 2003 – 2005 гг.;
- максимальный доход каждого года;
- количество кварталов матрицы с доходом выше  $\frac{1}{4}$  суммы доходов за 2003-2005 гг.;

- получить новую матрицу, в которой значения кварталов с доходом ниже  $\frac{1}{4}$  суммы доходов за 2003-2005 гг. заменить нулями.

–

### **Творческие задания**

*уровень начальный*

1. В матрице объема удоя молока фермерского хозяйства А (1-4 кв., 2000-2004 гг.) найти максимальное значение. Указать количество максимальных элементов, года и кварталы их встречи.
2. Для матрицы выпуска бытовой техники А (1-4 кв., 2002-2004 гг.) найти кварталы выпуска, значения которых больше среднего.

*уровень средний*

3. Для матрицы В (1-4 кв., 2001-2004гг.) найти разницу между минимальным и максимальным значениями. Указать год с минимальным значением.
4. В матрице спроса на ЭКСПРЕСС-кредит С (1-2 полугодия, 2000-2004 гг.) найти суммарное и максимальное значения. Указать процент максимального значения от суммарного.

*уровень высокий*

5. Для матрицы объема сбора урожая В (1-2 полугодия, 1995-2004гг.) найти средние значения XX(1995-2000 гг.) и XXI(2001-2004 гг.) и указать в каком веке наибольшее.
6. В матрице объема продаж мороженого С (1-2 полугодия, 2000-2004 гг.) найти среднеполугодовые значения S1 и S2. Получить новую матрицу как процент текущего полугодия к соответствующей сумме.
7. Для матрицы объема продаж авиабилетов А (1-4 кв., 2000-2002 гг.) получить новую матрицу как разницу между первым и текущим кварталами соответствующего года.

## **VIII. КОМПЛЕКТ ЗАДАНИЙ ДЛЯ КОНТРОЛЬНЫХ РАБОТ**

### **Контрольная работа по теме «Информатика. Информация».**

1. Данные—это
  - a) признаки или записанные наблюдения, которые не используются, а только хранятся;
  - b) связанные между собой сведения, понятия, изменяющие наши представления о явлении;
  - c) концептуально связанные характеристики предметной области, которые не используются.
2. Информация по месту возникновения может быть:
  - a) входная, выходная, внутренняя, внешняя;
  - b) плановая, нормативно-справочная, учетная, оперативная;
  - c) первичная, вторичная, промежуточная, результатная.
3. Информация—это
  - a) сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности;
  - b) отдельные документы и отдельные массивы документов, содержащие сведения об объектах окружающей среды;
  - c) данные, отражающие степень соответствия образа объекта и самого объекта, принадлежащего определенной предметной области.
4. Информационная система – это
  - a) совокупность средств и методов отбора, обработки и передачи данных для получения информации нового качества;
  - b) совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели;
  - c) совокупность сведений, отражающих происходящие процессы и служащие для управления этими процессами и коллективами людей.
5. По способу отображения информация бывает
  - a) текстовая, графическая;
  - b) числовая, текстовая, рисунки;
  - c) числа, формулы, текст, графика.
6. По признаку стабильности информация бывает
  - a) переменная, постоянная;
  - b) временная постоянная;
  - c) обрабатываемая, неизменяемая.

## **Контрольная работа по теме «Аппаратное обеспечение ПК»**

1. Системная шина—это
  - a) интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств,
  - b) устройство, имеющее провода и схемы сопряжения для подключения ПК к системе энергопитания;
  - c) устройство, подающее во все блоки ПК в нужные моменты времени управляющие сигналы.
2. Основная память состоит из:
  - a) КЭШ-памяти и BIOS;
  - b) ОЗУ и ПЗУ;
  - c) микропроцессорной памяти и адресного пространства.
3. Внешними запоминающими устройствами являются:
  - a) жесткий диск;
  - b) стример;
  - c) флоппи-диск.
4. Микропроцессор—это
  - a) блок ПК, предназначенный для управления работой всех блоков машины и для выполнения арифметических и логических операций над информацией;
  - b) система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой;
  - c) устройство, генерирующее последовательность электрических импульсов, и обеспечивающее согласованную работу всех блоков ПК.
5. ОЗУ предназначено для
  - a) кратковременного хранения информации в текущий момент времени;
  - b) ввода и работы с информацией в текущий момент времени;
  - c) длительного хранения неизменяемой информации.
6. Устройствами для вывода информации являются:
  - a) диджитайзеры;
  - b) плоттеры;
  - c) мониторы.

## **Контрольная работа по теме «Системы счисления»**

1. Наименьшей единицей информации является:
  - a) бит,
  - b) байт,
  - c) символ.
2. Информация в ЭВМ кодируется

- a) в двоичной системе счисления;
  - b) в десятичной системе счисления;
  - c) в восьмеричной системе счисления.
3. Машинное слово—это
- a) 2 байта;
  - b) 4 байта;
  - c) 1 байт.
4. В восьмеричной системе счисления базой являются числа
- a) от 0 до 7;
  - b) от 1 до 8;
  - c) от 0 до 8.
5. Байт – это
- a) 2 бита;
  - b) 4 бита;
  - c) 8 бит.
6. Системы счисления разделяют на
- a) двоичную и родственные ей;
  - b) римские и арабские;
  - c) позиционные и непозиционные.
7. 5120 байт это
- a) 5 килобайт;
  - b) 5.12 килобайт;
  - c) 0.512 килобайт.
8. Представьте число 12 в двоичной системе счисления:
- a) 1100;
  - b) 1101;
  - c) 1011.
9. Результатом суммы чисел, записанных в двоичной системе счисления, 101 и 1010 будет число
- a) 1111;
  - b) 10100;
  - c) 10011.

### **Контрольная работа по теме «Программное обеспечение ПК»**

1. Программа—это
- a) упорядоченная последовательность команд компьютера для решения поставленной задачи;
  - b) совокупность функций, с помощью которых достигается решение поставленной задачи;
  - c) система точно сформулированных правил, определяющая процесс решения задачи.
2. Архивация—это
- a) помещение исходных файлов в архив;
  - b) извлечение файлов из архива;
  - c) сохранение файлов.
3. Классы программных продуктов:
- a) системное программное обеспечение, пакеты прикладных программ, инструментарий технологии программирования;

- b) программы обслуживания сети, сервисное программное обеспечение, текстовые и графические редакторы;
  - c) базовое программное обеспечение, программные средства мультимедиа, системы искусственного интеллекта.
4. В классификации вирусов отсутствуют
- a) невидимки;
  - b) мутанты;
  - c) змеи.
5. В сервисное программное обеспечение входят:
- a) пакеты прикладных программ, операционные оболочки, программные средства мультимедиа, системы искусственного интеллекта, утилиты;
  - b) программы: диагностики работоспособности ПК, антивирусные, обслуживания дисков, архивирования данных, обслуживания сети;
  - c) СУБД, текстовые процессоры, табличные процессоры, интегрированные пакеты, коммуникационные системы.

### **Контрольная работа по теме «Компьютерная графика»**

1. Векторная графика: изображение
- a) состоит из контуров, которые представляют собой кривые имеющие точное математическое описание;
  - b) строится по математическому уравнению (не рисование, а программирование);
  - c) состоит из прямоугольной сетки точек.
2. Оптимальным средством создания и хранения чертежей, схем является:
- a) векторный графический редактор;
  - b) растровый графический редактор;
  - c) фрактальный графический редактор.

### **Контрольная работа по теме «Windows».**

#### **Теоретическая часть:**

#### **Вариант № 1**

1. Windows – это
- a) оболочка операционной системы;
  - b) драйвер;
  - c) операционная система.
2. Ярлык в Windows – это
- a) программа;
  - b) объект;
  - c) указатель на объект.

3. Укажите верное перечисление всех видов меню в Windows :
- a) главное меню, ниспадающее меню, системное меню, меню кнопки ПУСК;
  - b) контекстное меню, системное меню, строка меню, меню объекта;
  - c) системное меню, главное меню, строка меню, контекстное меню.
4. Что обозначают кнопки с изображением:
- a) свернуть окно на панель задач, изменить размеры окна, закрыть окно;
  - b) свернуть окно на панель задач, задать заголовок окна, закрыть окно;
  - c) закрыть окно, изменить размеры окна, открыть системное меню.
5. Проводник Windows предназначен для:
- a) для восстановления удаленной информации на компьютере;
  - b) для ввода текстовой информации;
  - c) для просмотра структуры папок и данных, и работы с ними.
6. Панелью инструментов называется:
- a) панель содержащая значки и кнопки, предназначенные для открытия ниспадающего меню;
  - b) панель содержащая значки и кнопки, предназначенные для быстрого доступа к наиболее часто используемым командам приложения;
  - c) панель содержащая значки и кнопки, предназначенные для использования команд отсутствующих во всех видах меню.
7. Для перемещения объекта необходимо выполнить следующие операции:
- a) выделить объект, выбрать команду "копировать", выбрать команду "вставить";
  - b) выделить объект, выбрать команду "забрать", выбрать команду "вставить";
  - c) выбрать команду "копировать", команду "забрать", команду "вставить".
8. Папка Windows—это:
- a) место для хранения программ, документов, дополнительных папок;
  - b) поименованная область на диске для хранения информации;
  - c) локальные сетевые диски и диски сетевых серверов.
9. Программа "Корзина" относится к:
- a) программам утилита;
  - b) системам программирования;
  - c) программам драйверам.
10. Имена файлов в Windows не должны:
- a) превышать 8 символов;
  - b) содержать буквы русского алфавита;
  - c) превышать 255 символов.

## **Практическая часть**

1. На диске С создайте папку – в качестве имени папки введите свою фамилию.
2. В этой папке создайте 2 папки ПРИЕМНИК и ИСТОК.
3. Создайте в папке ИСТОК два файла: ярлык.doc (дайте определение ярлыка) и файл.doc (дайте определение файла).
4. Скопируйте ярлык.doc в папку ПРИЕМНИК.
5. Создайте файл Poisk.doc (напишите для чего предназначена программа «Поиск») в папке ФИО.
6. Скопируйте папку ИСТОК на диск С.
7. Переименуйте папку ИСТОК, находящуюся на диске С.
8. Переместите файл.doc из папки ИСТОК на Рабочий стол.
9. Создайте ярлык для файла Poisk.doc.
10. Удалите папку ПРИЕМНИК, поместив ее в Корзину.
11. Восстановите удаленную папку.
12. Смените значок ярлыка.
13. Осуществите быстрый поиск файлов:
  - Все файлы, начинающиеся с буквы О.
  - Все файлы, имя которых не больше 5 символов, с расширением txt.
  - Графический файл, начинающийся на букву Т.

## **Контрольная работа по теме «MS DOS»**

### **Теоретическая часть**

#### **Вариант №1.**

1. MS DOS – это
  - a) операционная система;
  - b) оболочка операционной системы;
  - c) утилита.
2. Путь задает маршрут к тому каталогу, где находится файл. Путь - это
  - a) последовательность имен каталогов, разделенных символом "\";

- b) последовательность имен каталогов, разделенных символом " /";
  - c) последовательность имен каталогов, разделенных символом " +".
3. Файл—это
- a) поименованная область на диске;
  - b) часть диска;
  - c) текст или изображение.
4. Каталог—это
- a) место хранения имен файлов;
  - b) структура диска;
  - c) область в оперативной памяти.
5. Шаблон имени файла—это форма, в которой в полях имени и типа файла используются символы
- a) + и ?
  - b) # и &
  - c) и ?
6. В MS DOS для просмотра каталога используется команда
- a) MD;
  - b) TYPE;
  - c) DIR.
7. Укажите команду переименования файла abc.txt в mmm.doc:
- a) move abc.txt mmm.doc
  - b) copy abc.txt mmm.doc
  - c) ren abc.txt mmm.doc
8. Укажите команду перехода из текущего каталога DAR в каталог MIF на диске A:
- a) cd A:MIF
  - b) rd A:MIF
  - c) md A:MIF

### **Практическая часть**

1. Создайте текстовый файл **txt1.txt**, содержащий следующий текст:  
*Файл - это поименованная область на магнитном носителе.*
2. Создайте текстовый файл **txt2.txt**, содержащий следующий текст:  
*Корневой каталог — это главная директория диска, главный ящик, в котором лежат все остальные ящички - директории.*
3. Скопируйте содержимое файлов **txt1.txt** и **txt2.txt** в новый файл с именем file.txt.

4. Выведите содержимое файла **file.txt** на экран.
5. Переименуйте файл **file.txt** в файл **text.doc**.
6. Создайте текстовый файл **fax.bak**:  
*Панель задач — позволяет легко получить доступ ко всем открытым и работающим программам.*
7. Создайте текстовый файл **fff.bak**:  
*Общение с операционной системой MS DOS 6.0-6.22.*
8. Выполните копирование файла **text.doc** с диска C: на диск A.
9. Выполните групповое копирование файлов, имена которых начинаются с буквы f, с диска C: на диск A.
10. Выведите оглавление каталога диска A: сколько файлов находится на диске A? 11
11. Создайте на диске A: каталог **PROBA**. Убедитесь, что каталог создан.
12. Скопируйте из корневого каталога в каталог **PROBA** файл text.doc.
13. Выведите на экран оглавление каталога **PROBA**.
14. Показать преподавателю проделанную работу.

### **Контрольная работа по теме «Total Commander»**

#### **Теоретическая часть**

#### **Вариант №1.**

1. Total Commander - это:
  - a) интегрированный пакет прикладных программ;
  - b) сервисная программа;
  - c) системная оболочка пользователя.
  - d) операционная система
2. Помощь вызывается нажатием клавиши:
 

a) Alt+F1;	c) Ctrl+F1;
b) F1;	d) Shift+F1.
3. Назначение клавиши F5 состоит:
  - a) в копировании файла;
  - b) в создании каталога;
  - c) в удалении файла;
  - d) в перемещении файла
4. Редактирование текстового файла в TC осуществляется клавишами:
 

a) F6;	c) F3;
b) F4;	d) F2

6. Нажатие клавиш Alt+F2 в ТС служит для:
- a) гашения/восстановления правой панели;
  - b) перехода на другой диск в правой панели;
  - c) перехода на другой диск в левой панели.
  - d) гашения/восстановления левой панели;
7. С помощью клавиши Alt+F7 можно:
- a) создать файл;
  - b) поиск файлов;
  - c) просмотреть файл.
  - d) Редактировать файл
8. С помощью клавиши F7 можно:
- a) создать файл;
  - b) создать каталог;
  - c) просмотреть файл.
  - d) просмотреть каталог
9. Поиск файла осуществляется с помощью клавиши:
- a) Ctrl+F8;
  - b) Alt+F8;
  - c) Alt+F7.
  - d) Ctrl+F7
10. Допускаются следующие режимы сортировки файлов (укажите 3 правильные ответа):
- a) по размеру;
  - b) по расширению;
  - c) без сортировки;
  - d) по содержанию;
  - e) по количеству содержащихся в нем объектов.

### **Практическая часть**

1. В корневом каталоге диска C: создайте каталог ORBIT A.
2. В каталоге ORBITA создайте текстовый файл paloma.txt
3. Скопировать файл paloma.txt в файл stih.txt.
4. Создайте каталог DATA.
5. Перенесите каталог DATA в ORBITA.
6. Создайте текстовый файл rok.txt в каталоге DATA.
7. Переименуйте текстовый файл rok.txt в файл wad.txt.
8. Работая, с управляющим меню выполните:
  - Выведите дерево каталогов на неактивной панели;
  - Выведите информацию о диске.
9. Осуществите на диске поиск файлов: \*.doc, d\*.\*?t

## Контрольная работа по теме «Word»

### Теоретическая часть:

#### Вариант № 1

- 1) Шрифт-это...
  - a) набор специальных символов русского и английского алфавитов, отображаемых определенным образом;
  - b) набор букв и цифр русского алфавита;
  - c) набор букв, цифр, специальных символов, знаков препинания, отображаемых определенным образом;
  - d) выделенная отличным цветом, начертанием, анимацией, часть текста.
- 2) Колонтитул-это...
  - a) граница текста;
  - b) вставленный графический объект;
  - c) строки текста, печатаемые на всех полях страницы;
  - d) строки текста, печатаемые на верхних и нижних полях страницы.
- 3) Для организации таблицы следует выполнить команду ... и указать число строк, столбцов
  - a) Вставка →Таблица;
  - b) Таблица→Нарисовать;
  - c) Таблица→Добавить;
  - d) Формат→Таблица.
- 4) Деления горизонтальной линейки могут отображаться в ... Уберите лишний вариант
  - a) см;
  - b) мм;
  - c) пунктах;
  - d) дюймах;
- 5) На панели форматирования находятся →кнопки:
  - a) открыть, сохранить, просмотреть;
  - b) вырезать, копировать, вставить;
  - c) жирный, курсивный, подчеркнутый;
  - d) цвет линии, цвет заливки, цвет шрифта.
- 6) Выделение одного символа
  - a) Shift+курсор;
  - b) Ctrl+ курсор;
  - c) Alt+курсор;
  - d) Ctrl+ Shift + курсор.
- 7) Для подсчета среднего арифметического в таблице Word используется формула:
  - a) SUM;
  - b) LEFT;
  - c) AVERAGE;
  - d) PRODUCT.
- 8) Копирование выделенного абзаца выполняется так:
  - a) Правка→Копировать;

- b) Формат→Абзац→Копировать;
  - c) Абзац→Копировать;
  - d) Файл→Копировать.
- 9) Заменить одно слово на другое можно командой:
- a) Сервис→Правописание→Заменить;
  - b) Правка→Правописание→Заменить;
  - c) Правка→Заменить;
  - d) Сервис→Заменить.
- 10) Для ввода степени числа используют:
- a) шрифт маленького размера;
  - b) подходящий по размеру формат абзаца;
  - c) вставку специального символа;
  - d) верхний индекс.

### **Практическая часть**

1. Посмотреть имеющиеся шаблоны.
2. Создать шаблон листка текущей аттестации.
3. Установить альбомную ориентацию.
4. Создать газетный листок
  - заголовок газеты разместить по центру, разрядку, подбор шрифта и размера, обрамление сверху и снизу задать произвольно;
  - установить № выпуска и дату;
  - используйте буквицу.
  - перечень рубрик сформатировать при помощи таблицы и маркированного списка;
  - вставить конец раздела;
  - оставшуюся часть листа разбить на три колонки;
5. Создайте новый документ
6. Откройте окно Справки. Скопируйте какой-либо абзац из нее в документ.
7. Размножьте его три раза.
8. Примените к двум абзацам имеющиеся стили.
9. Создайте свой стиль и примените его к третьему абзацу.
10. Покажите результат преподавателю.
11. Закройте Word.

## Контрольная работа по теме «EXCEL»

### Теоретическая часть:

#### Вариант № 1

- 1 Содержимым отдельной ячейки может быть:
  - a) число, текст, или формула;
  - b) график, диаграмма или рисунок;
  - c) команда, файл, каталог.
2. Для задания диапазона ячеек нужно указать:
  - a) его левый верхний и правый нижний адреса ячеек;
  - b) адрес каждой ячейки диапазона;
  - c) адреса двух ячеек с максимальным и минимальным элементами.
3. Для задания диапазона ячеек два требуемых адреса необходимо разделить:
  - a) двоеточием;
  - b) тире;
  - c) многоточием (три точки).
4. При копировании из ячейки-источника, содержащей адрес какой-либо ячейки, в ячейку-приемник, находящуюся ниже, в ячейке-приемнике происходит изменение содержимого ячейки-источника:
  - a) изменяется индекс строки;
  - b) изменяется индекс столбца;
  - c) автоматически перед требуемым индексом появляется знак доллара.
5. При копировании из ячейки-источника, содержащей адрес какой-либо ячейки, в ячейку-приемник, находящуюся правее, в ячейке-приемнике происходит изменение содержимого ячейки-источника:
  - a) изменяется индекс столбца;
  - b) изменяется индекс строки;
  - c) автоматически перед требуемым индексом появляется знак доллара.
6. Вопрос: Для предотвращения изменения индексов (при копировании формулы в адресном виде) из ячейки-источника в ячейку-приемник необходимо:
  - a) предварительно перед индексами строк и столбцов поставить знак доллара;
  - b) предварительно после индексов строк и столбцов поставить знак доллара;
  - c) в процессе копирования нажать и не отпускать функциональную клавишу Ctrl
7. При сортировке списка, содержащего несколько столбцов необходимо указать:
  - a) диапазон всего списка;
  - b) диапазон ключевого столбца;

с) диапазон ячеек с заголовками столбцов.

### Практическая часть

Подготовьте и заполните таблицу:

ФИО	Профессия	разряд	Числа месяца					Дни явок	Дни неявок			Отработано часов
			1	2	...	30	31		Отпуск	Болезнь	Пргул	

1. Проставьте для каждого сотрудника:

- количество часов, отработанных за день, или
- о, если он находился в отпуске, или
- б, если в этот день сотрудник болеет, или
- п, если он прогуливает.

2. Введите формулу для подсчета количества дней, пропущенных по болезни (функция СЧЕТЕСЛИ, критерий «б»).

3. В отдельной ячейке введите формулу для подсчета количества нерабочих дней месяца. Вставьте для этого функцию СЧИТАТЬ ПУСТОТЫ, т.к. в таблице в такие дни ничего не отмечено.

4. Введите формулу для подсчета прогулов (функция СЧЕТЕСЛИ, критерий «п»).

5. Постройте плоскую гистограмму, в которой будут отражены дни явок и число отработанных часов для каждого сотрудника.

### Контрольная работа по теме «ACCESS»

#### Теоретическая часть

#### Вариант № 1

1) Ключом записи таблицы «Студент» реляционной БД может быть поле, содержащее следующие данные:

- а) номер группы;
- б) факультет;
- с) номер контракта;
- д) изучаемая дисциплина.

2) Системы управления базами данных, ориентированные на ПК, как правило, поддерживают модель данных:

- а) иерархическую;
- б) реляционную;
- с) сетевую;
- д) семантическую.

3) Установленные связи между таблицами реляционной базы данных помогают:

- a) избежать дублирования информации;
  - b) определить местонахождение нужной таблицы;
  - c) производить сортировку таблицы;
  - d) распечатать данные таблицы.
- 4) Система управления базами данных - это:
- a) формальный аппарат ограничений на формирование таблиц, который позволяет устранить дублирование;
  - b) комплекс программных и языковых средств, предназначенных для создания, ведения и совместного применения баз данных многими пользователями;
  - c) совокупность четко определенных целенаправленных действий персонала;
  - d) система, реализующая сбор, обработку и манипулирование данными и включающая технические средства, программное обеспечение и соответствующий персонал.
- 5) В настоящее время наиболее распространенные СУБД:
- a) иерархические;
  - b) сетевые;
  - c) реляционные;
  - d) смешанные.
- 6) Запись в БД ACCESS:
- a) столбцы реляционной таблицы;
  - b) строки реляционной таблицы;
  - c) одно или несколько полей, имеющих уникальное значение для реляционной таблицы;
  - d) отчет, состоящий только из одной строки.

### **Практическая часть**

Выберете самостоятельно тему и создайте БД, выполнив следующие условия:

1. создайте таблицу.
2. Внесите записи.
3. Отсортируйте их.
4. Создайте форму.
5. Внесите еще 2 записи в форму.
6. Организуйте поиск, задав критерий для поиска.
7. Представить форму в режиме таблицы.
8. Примените фильтр, задав условие отбора данных.
9. Подготовьте отчет.

## Контрольная работа по «PowerPoint»

### Вариант №1

#### Теоретическая часть

1. Основной элемент презентации:

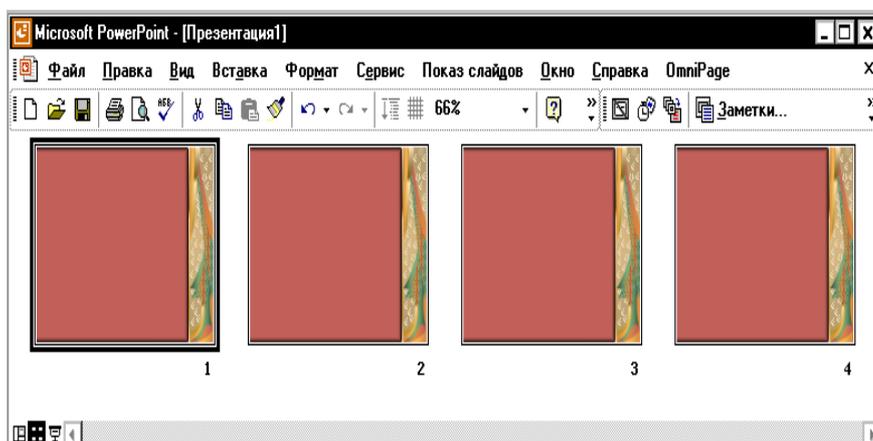
1. рисунок;
2. страница;
3. слайд.

2. Презентацию можно сохранить с расширением

1. .ppt .pps
2. .ppt .dbt
3. .ptp .spp

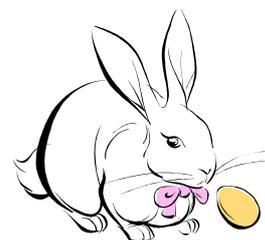
3. В каком режиме просматривается данная презентация?

1. обычный;
2. страницы заметок;
3. сортировщик слайдов.



4. Укажите способ вставки рисунка из коллекции MS Office в презентацию

1. Вставка→Рисунок→Картинки;
2. Сервис→Рисунок→Картинки;
3. Вид→Вставка→Вставка картинок.



5. Для выбора фона слайда используется команда:

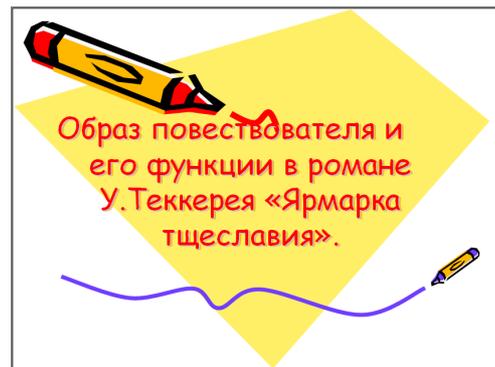
1. Формат→Фон;
2. Формат→Оформление слайда...;
3. Вид→Фон.

6. Добавить новый слайд в презентацию можно с помощью

1. Файл→Создать;
2. Вставка→Создать слайд;
3. Формат→Новый слайд.

7. Укажите тип выравнивания текста, приведенного на рисунке:

1. по центру;
2. по ширине;
3. по правому краю.

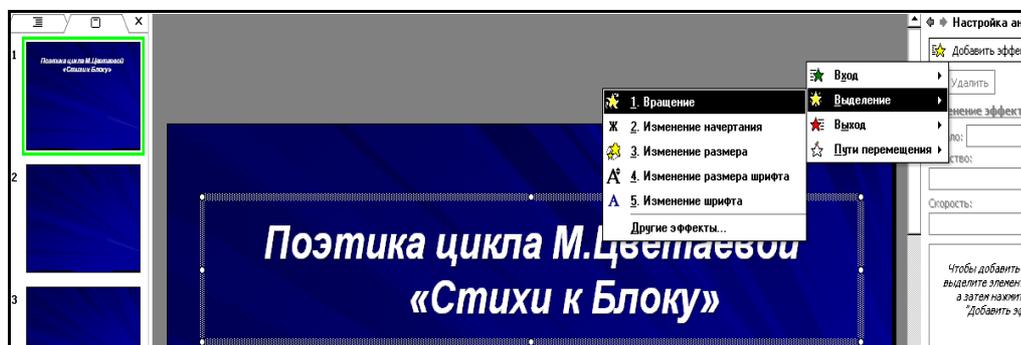


8. Для того чтобы добавить переходы к слайду нужно:

1. выделить слайд, затем щелкнуть Показ слайдов→Смена слайдов и выбрать переход;
2. выделить слайд, затем щелкнуть Вставка→Добавить переход и выбрать переход;
3. выделить слайд, затем щелкнуть Формат→Переходы и выбрать переход.

9. На рисунке изображено окно Настройки анимации. Выделенный черным цветом эффект Вращение будет применен к

1. выделенному зеленым цветом слайду;
2. выделенному на слайде объекту;
3. всем слайдам.



10. Можно ли изменить цветовую схему шаблона оформления слайда?

1. Да: команда Формат→Заливка слайда;
2. Нет;
3. Да: команда Формат→Фон.

## Контрольная работа по теме «HTML»

### Теоретическая часть

#### Вариант №1

- Код программы на языке HTML—это
  - обычный текстовый файл, созданный в текстовом редакторе;
  - программа, написанная на специальном языке программирования;
  - программа, написанная в Internet Explorer.
- Все HTML–теги пишутся в скобках:
  - <...>;
  - (...);
  - /.../.
- Тег FONT позволяет управлять
  - размером и цветом шрифта;
  - начертанием шрифта;
  - цветом фона.
- Отдельный абзац задается с помощью тега
  - H;
  - B;
  - P.
- Для создания нумерованного списка используют тег
  - OL;
  - UL;
  - DL.
- Для вставки графики используется тег
  - IMG;
  - GMI;
  - IGM.
- С помощью каких тегов описывается таблица:
  - <TABLE>
  - <BODY>
  - <TD>
- Гипертекстовые ссылки задаются тегами:
  - <A>...</A>
  - <P>...</P>
  - <B>...</B>т
- Фреймовая структура задается с помощью тега:
  - FRAMESET;
  - FRAME;
  - FRAMEBORDER.
- Каждый фрейм создается как
  - элемент веб-страницы;
  - отдельная веб-страница;
  - специальная программа.

### Практическая часть

1.Создайте веб-страницу. (Цвет фона выберите сами.) На ней разместите:

- Заголовок страницы «Контрольная работа»
- Заголовок документа «Компьютерные вирусы» отделите от текста линией.
- Текст документа напишите курсивом и отформатируйте по центру.

Текст:

*Компьютерный вирус – это специально написанная программа, способная самопроизвольно присоединяться к другим программам, создавать свои копии и внедрять их в файлы, системные области компьютера, вычислительные сети с целью нарушения работы программ, порчи файлов: и каталогов, создания всевозможных помех в работе на компьютере.*

4. Маркированный список:

Классифицировать вирусы можно по следующим признакам:

- Среде обитания;
- Способу заражения среды обитания
- Воздействию;
- Особенности алгоритма.

5. Таблица:

Вирусы	
Среда обитания	Сетевые Файловые Загрузочные Файлово-загрузочные
Способ заражения	Резидентные Нерезидентные
Степень воздействия	Неопасные Опасные Очень опасные
Особенности алгоритмов	Паразитические Черви Невидимки Мутанты Троянские

II. Создайте веб-страницу с двумя фреймами. В одном фрейме разместите картинку, в другом текст (придумайте сами).

## Контрольная работа по теме «MathCAD»

### Теоретическая часть

#### Вариант №1

- Документ MathCAD состоит из блоков. Выполнение блоков происходит:
  - слева-направо и сверху-вниз;
  - справа-налево и сверху-вниз;
  - согласно их нумерации.
- Для включения/выключения панелей инструментов используется пункт меню:
  - Format;
  - View;
  - Symbolic.
- Для присваивания значения переменной используется знак:
  - =
  - :=
  - =
- Кнопка  открывает:
  - панель символьных операций;
  - панель греческих букв;
  - панель программирования.
- $x$  принимает значения от -2 до 2 с шагом 0.1. На языке MathCAD это будет записано так:
  - $x := -2, 0.1..2$
  - $x = -2, 0.1..2$
  - $x := -2, 1.9..2$
- Для того, чтобы решить уравнение используют функцию:
  - factor;
  - expand;
  - solve.
- Дана система уравнений:
$$\begin{cases} 3y - 5x = 1, \\ 2x + 3y = 5. \end{cases}$$
 Укажите правильную запись решения в

MathCAD:

a)

Given  
 $3 \cdot y - 5 \cdot x = 1$   
 $2 \cdot x + 3 \cdot y = 5$

b)

Given  
 $3 \cdot y - 5 \cdot x = 1$   
 $2 \cdot x + 3 \cdot y = 5$

c)

Given  
 $3 \cdot y - 5 \cdot x = 1$   
 $2 \cdot x - 3 \cdot y = 5$

- Дано:  $f(x) = x^2 + 2x + 1$ ,  $x$  изменяется от -5 до 5 с шагом 0.1. Построить таблицу значений  $f(x)$ . Укажите правильную запись решения в MathCAD:

a)  $f(x) := x^2 + 2 \cdot x + 1$   
 $x := -5, -4.9..5$   
 $f(x) =$

b)  $f(x) := x^2 + 2 \cdot x + 1$   
 $x = -5, 0.1..5$   
 $f(x)$

c)  $f(x) := x^2 + 2 \cdot x + 1$   
 $x := -5, 0.1..5$   
 $f(x) =$

<ENTER>

9. Найти:  $\lim_{x \rightarrow 0} \frac{\sin 5x}{x}$ . Укажите верную запись решения:

a)  $\lim_{x \rightarrow 0} \frac{\sin 5x}{x} \rightarrow$

b)  $\lim_{x \rightarrow 0} \frac{\sin 5x}{x} =$

c)  $\lim_{x \rightarrow 0} \frac{\sin 5x}{x}$  solve,  $x \rightarrow$

10. Для того чтобы выполнить какую-либо операцию с матрицей нужно:

a) Выделить матрицу, щелкнуть в панели по кнопке операции, затем поставить знак =;

b) Выделить матрицу, щелкнуть в панели по кнопке операции, затем щелкнуть вне формулы;

c) Выделить матрицу, щелкнуть в панели по кнопке операции, затем нажать <ENTER>.

### Практическая часть

1. Вычислите значение выражения:  $\left(\frac{25}{9}\right)^3 + \sqrt{6}$ .

2. Постройте график функции  $y(x) = \frac{1}{x^2 + 1}$  на промежутке  $[-5, 5]$ .

3. Вычислите неопределенный и определенный интегралы:  $\int \frac{x+2}{4x^2+5x+1} dx$ ,

$$\int_0^1 x \sqrt{x^2+1} dx .$$

4. Упростите выражение:  $\frac{2}{3} \left( \frac{1+3x}{1-3x} \right) \left( \frac{-6x}{(1+3x)^2} \right)$ .

5. Разложите на множители:  $a^4 - b^4$ .

6. Решите систему уравнений 
$$\begin{cases} x + y^2 = 3, \\ y^2 - 3x = -5. \end{cases}$$

7. Дано:  $A = \begin{pmatrix} 3 & 2 & 2 \\ 1 & 3 & 1 \\ 5 & 3 & 4 \end{pmatrix}$ . Вычислите определитель матрицы  $A$ , найдите матрицу

обратную по отношению к матрице  $A$ , возведите матрицу  $A$  в квадрат.

8. Вычислите предел:  $\lim_{x \rightarrow 5} \frac{x^2 + 2}{x - 3}$ .

9. Найдите производную первого порядка функции:  $y(x) = \sqrt{2x+1} + \ln(x^2 + 5x)$ .

## Контрольные работы по теме «Компьютерные сети. Интернет.»

1. Компьютерная сеть—это
  - a) совокупность компьютеров и терминалов, соединенных каналами связи в единую систему;
  - b) группа компьютеров выполняющих единый информационно-вычислительный процесс;
  - c) независимые, но связанные между собой компьютеры, выполняющие обработку данных.
2. Типовые топологии ЛВС:
  - a) кольцо, звезда, шина;
  - b) кольцо, снежинка, паутина;
  - c) кольцо, паутина, звезда.
3. Протокол—это
  - a) набор правил, определяющий взаимодействие двух систем;
  - b) программа, определяющая взаимодействие между системами;
  - c) некоторая последовательность действий для достижения цели.
4. Протокол, обеспечивающий передачу файлов между двумя компьютерами, подключенными к сети Internet:
  - a) FTP;
  - b) TFP;
  - c) FTT.
5. В зависимости от территориального расположения абонентских систем вычислительные сети можно разделить на:
  - a) глобальные, региональные, локальные;
  - b) цифровые, аналоговые, широкополосные;
  - c) локальные, дистанционные, многошинные.
6. Языками создания гипертекстовых документов являются:
  - a) HTML;
  - b) Java;
  - c) Prolog.

## Контрольные работы по теме «Delphi»

**Тема: Знакомство с системой программирования Delphi**

### Вариант №1

1. Delphi реализует интерфейс управления окнами:
  - a) SDI (Single Document Interfase);
  - b) MDI (Multiple Document Interface);
  - c) RDI (Reformative DI).
2. Окно формы используется для:

- a) проектирования окна программы;
  - b) выбора визуальных компонентов;
  - c) создания текста программы.
- 3. В инспекторе объектов задаются:**
- a) заголовок формы;                      b) свойства и события;                      c) программа и алгоритм.
- 4. Страница Properties используется:**
- a) для задания свойств объекта;
  - b) для задания событий объекта";
  - c) в меню главного окна.
- 5. TButton является:**
- a) командной кнопкой;                      b) меткой;                      c) свойством.
- 6. Свойство Font определяет:**
- a) параметры шрифта;
  - b) палитру фона;
  - c) характеристики формы.
- 7. Свойства размера объекта:**
- a) Width, Height;                      b) Width, Top;                      c) Left, Top.
- 8. TButton не имеет свойства:**
- a) Caption;                      b) Color;                      c) Left.
- 9. Событие щелчок мыши:**
- a) OnClick;                      b) OnCreate;                      c) OnClose.
- 10. Верное описание события:**
- a) Procedure Form1. Button1Click();
  - b) Procedure Form1.OnClick(Button1);
  - c) Procedure Button1.Click().
- 11. Команда прекращения программы:**
- a) Project/Compile;                      b) Run/Run;                      c) Run/Reset.
- 12. Выполняемый файл \*.exe создается:**
- a) при компиляции сохраненной программы;
  - b) при алгоритмизации программы;
  - c) во время сохранения программы

## Тема: Программирование вычислительных задач

### Вариант №1

**1. Этап алгоритмизация следует после:**

- a) этапа программирования;
- b) этапа тестирования;
- c) этапа методологии решения.

**2. Компонент TEdit представляет:**

- a) окно выбора события;
- b) строку сообщения;
- c) однострочное окно ввода-вывода.

**3. Функция FloatToStr переводит:**

- a) целое число в строку;
- b) строку в вещественное число;
- c) вещественное число в строку.

**4. Переменные описывают в разделе:**

- a) var;
- b) begin;
- c) type.

**5. Выберите верную команду:**

- a) `S:=FloatToInt(edit1.text);`
- b) `S:=FloatToStr(edit1.text);`
- c) `S:=StrToInt(edit1.text);`

**6. Функция возведение  $x$  в квадрат:**

- a) `SQRT(x);`
- b) `SQR(x);`
- c) не существует.

**7. Функция  $(r \bmod 2)$  находит:**

- a) модуль числа;
- b) целую часть от деления;
- c) остаток от деления.

**8.  $ABS((5-x)+sqr(x))$  при  $x=9$  равно:**

- a) -1;
- b) 77;
- c) 19.

**9. Этап отладки осуществляется:**

- a) после анализа результатов тестирования;
- b) перед тестированием программы;
- c) на этапе алгоритмизации.

**10. Ошибки программы бывают:**

- a) структурные и алгоритмические;
- b) синтаксические и структурные;
- c) синтаксические и орфографические.

## 11. Структура программной единицы:

- a) заголовок, описание переменных, описание событий;
- b) заголовок, начало и окончание программы;
- c) заголовок, описания данных, раздел операторов.

## Тема: Программы с ветвлением. Компонент TCheckBox

### Вариант №1

1. Ветвление бывает:

- a) полное/неполное/частичное;
- b) однозначное/многозначное;
- c) полное/ неполное.

2. Краткая форма оператора IF:

- a) If <условие> else <оператор>;
- b) If <условие> then <оператор1> else <оператор2>;
- c) If <условие> then <оператор>.

3. Для связи условий используются:

- a) арифметических операций;
- b) логических операций;
- c) операций сравнения.

4. Условие  $(x \geq 5)$  or  $(x < -10)$  при  $x = -20$  равно:

- a) boolean;
- b) false;
- c) true.



5. :

- a) вариантного ветвления;
- b) полного ветвления;
- c) неполного ветвления.

6. В Pascal условие принадлежности промежутку :

- a)  $(-5 < x \leq 7)$ ;
- b)  $(x > -5)$  or  $(x \leq 7)$ ;
- c)  $(x > -5)$  and  $(x \leq 7)$ .

7. Свойство, включающее флажок:

- a) Caption;
- b) Color;
- c) State.

8. У TCheckBox отключают состояние:

- a) Нет;
- b) Не знаю;
- c) Ложь.

9. If  $(x < 4)$  and  $(x > -2)$  then  $y := x + 2$  else  $y := x - 5$ ; при  $x = 5$  вернет значение:



- a) ListBox1.ItemIndex(0);
- b) ListBox1.Items:=1;
- c) ListBox1.ItemIndex:=0.

**10. Свойство список строк TRadioGroup:**

- a) Lines;
- b) ItemIndex;
- c) Items.

**11. Радиокнопка называется:**

- a) автоматический переключатель;
- b) независимый переключатель;
- c) зависимый переключатель.

**12. Количество строк в RadioGroup1:**

- a) n:=RadioGroup1.Items.Count;
- b) n:=RadioGroup1.Count;
- RadioGroup1.ItemIndex.Count(n).

**Тема: Программы циклической структуры**

**Вариант №1**

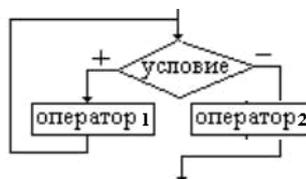
**1. Циклический оператор содержит:**

- a) константу выбора;
- b) две и более цепочки действий;
- c) повторяющиеся команды.

**2. Условие проверяется в начале цикла:**

- a) Repeat ...;
- b) во всех циклических операторах;
- c) While ....

**3. Выберите верное утверждение:**



- a) блок-схема цикла с предусловием;
- b) TRUE, FALSE нужно поменять местами;
- c) оператор2 веточки FALSE лишний.

**4. Цикл с предусловием имеет вид:**

- a) Repeat (<условие>) do <оператор>;
- b) While (<условие>) then <оператор>;

c) While (<условие>) do <оператор>.

5. Выберите верное утверждение:



- a) блок-схема цикла с постусловием;
- b) TRUE, FALSE нужно поменять местами;
- c) отсутствует второй оператор.

6. Параметр цикла с постусловием:

- a) обязателен и изменяется автоматически;
- b) задается и изменяется программистом;
- c) изменяется каждый раз только на один.

7. Тело цикла с параметром содержит:

- a) любое количество операторов;
- b) только один оператор;
- c) не более двух операторов.

8. a:=3; S:=0; For i:=2 to 5 do S:=S+a;

- a) результат S=15;
- b) результат S=12;
- c) результат S=S+a.

9. Начальное значение произведения:

- a) 1; б) -1; в) 0.

10. Отличие циклов с параметром и постусловием:

- a) параметр цикла изменяется автоматически/оператором;
- b) тело цикла выполняется n/(n-1) раз;
- c) условие записывается в одну/две строки.

**Тема: Программы работы с векторами**

**Вариант №1**

**1. Тип данных массив относится к:**

- a) структурированным типам данных;
- b) простым типам данных;
- c) объектным типам данных.

**2. Вектор объема продаж квартир A (2000-2004 гг.) описывается:**

- a) A: array [2000-2004] of Integer;

b) A: array [2000..2004] of real;

c) A:=IntToStr ([2000,2004]).

**3.Обращение к 2001 г. вектора A задания 2 записывается:**

a) A(1);

b) A[2001];

c) A(2).

**4.For i:=2000 to 2002 do S:=S+A[i]:**

a) находится сумму значений за три года;

b) находится количество значений;

c) находится среднегодовое значение.

**5.for i:=2000 to 2004 do**

**edit11.text:=IntToStr(i):**

a) выведет на экран номер года;

b) выведет на экран объем продаж i-года;

c) выведет количество лет вектора.

**6.k:=5; A[2000]:=14; A[2001]:=12;**

**A[2002]:=13; A[2003]:=16; A[2004]:=17;**

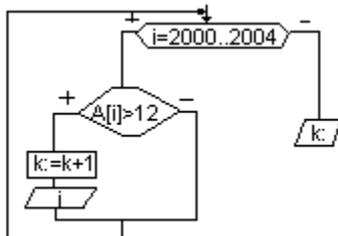
**For i:=2000 to 2004 do**

**If A[i]>14 then k:=k-1;**

**Выберите конечное значение k:**

a) k = 1; b) k = 3; c) k = 2.

**7.Выберете верное утверждение:**

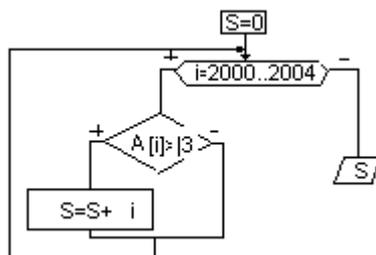


a) K выводится после выполнения цикла;

b) K выводится при ложности ветвления;

c) K выводится при истинности ветвления.

**8.Согласно вопросу 6 S равно:**



a) S=47; b) S=3; c) S=6007

## IX. ЗАДАНИЯ ДЛЯ ЗАЧЕТА (040201)

### 1 вариант

1. Данные—это
  - a) признаки или записанные наблюдения, которые не используются, а только хранятся;
  - b) связанные между собой сведения, понятия, изменяющие наши представления о явлении;
  - c) концептуально связанные характеристики предметной области, которые не используются.
2. Информация по месту возникновения может быть:
  - a) входная, выходная, внутренняя, внешняя;
  - b) плановая, нормативно-справочная, учетная, оперативная;
  - c) первичная, вторичная, промежуточная, результатная.
3. Наименьшей единицей информации является:
  - a) бит,
  - b) байт,
  - c) символ.
4. Системная шина—это
  - a) интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств,
  - b) устройство, имеющее провода и схемы сопряжения для подключения ПК к системе энергопитания;
  - c) устройство, подающее во все блоки ПК в нужные моменты времени управляющие сигналы.
5. Основная память состоит из:
  - a) КЭШ-памяти и BIOS;
  - b) ОЗУ и ПЗУ;
  - c) микропроцессорной памяти и адресного пространства.
6. Внешними запоминающими устройствами являются:
  - a) жесткий диск;
  - b) стример;
  - c) флоппи-диск.
7. Классификация ЭВМ по размерам и функциональным возможностям:
  - a) суперЭВМ, большие ЭВМ, малые ЭВМ, микроЭВМ;
  - b) универсальные ЭВМ, большие ЭВМ, средние ЭВМ, малые ЭВМ;
  - c) суперЭВМ, большие ЭВМ, гибридные ЭВМ, малые ЭВМ, универсальные ЭВМ.
8. Компьютерная сеть—это
  - d) совокупность компьютеров и терминалов, соединенных каналами связи в единую систему;

- е) группа компьютеров выполняющих единый информационно-вычислительный процесс;
  - ф) независимые, но связанные между собой компьютеры, выполняющие обработку данных.
9. Типовые топологии ЛВС:
- д) кольцо, звезда, шина;
  - е) кольцо, снежинка, паутина;
  - ф) кольцо, паутина, звезда.
10. Протокол, обеспечивающий передачу файлов между двумя компьютерами, подключенными к сети Internet:
- д) FTP;
  - е) TFP;
  - ф) FTT.
11. Программа—это
- а) упорядоченная последовательность команд компьютера для решения поставленной задачи;
  - б) совокупность функций, с помощью которых достигается решение поставленной задачи;
  - с) система точно сформулированных правил, определяющая процесс решения задачи.
12. Признаком появления вирусов не является:
- а) неправильная работа программ;
  - б) увеличение размеров файлов;
  - с) мерцание монитора.
13. В сервисное программное обеспечение входят:
- д) пакеты прикладных программ, операционные оболочки, программные средства мультимедиа, системы искусственного интеллекта, утилиты;
  - е) программы: диагностики работоспособности ПК, антивирусные, обслуживания дисков, архивирования данных, обслуживания сети;
  - ф) СУБД, текстовые процессоры, табличные процессоры, интегрированные пакеты, коммуникационные системы.
14. Каталог—это
- а) место хранения имен файлов;
  - б) часть диска;
  - с) область в оперативной памяти.
15. Программа «Проводник» предназначена для
- а) управления файловой системой и обеспечивает доступ к локальным и сетевым ресурсам;



- d) состоит из контуров, которые представляют собой кривые имеющие точное математическое описание;
- e) строится по математическому уравнению (не рисование, а программирование);
- f) состоит из прямоугольной сетки точек.

## 2 вариант

### 1. Информация—это

- a) сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности;
- b) отдельные документы и отдельные массивы документов, содержащие сведения об объектах окружающей среды;
- c) данные, отражающие степень соответствия образа объекта и самого объекта, принадлежащего определенной предметной области.

### 2. Информация в ЭВМ кодируется

- a) в двоичной системе счисления;
- b) в десятичной системе счисления;
- c) в шестнадцатеричной системе счисления.

### 3. Машинное слово—это

- a) 2 байта;
- b) 4 байта;
- c) 1 байт.

### 4. Микропроцессор—это

- a) блок ПК, предназначенный для управления работой всех блоков машины и для выполнения арифметических и логических операций над информацией;
- b) система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой;
- c) устройство, генерирующее последовательность электрических импульсов, и обеспечивающее согласованную работу всех блоков ПК.

### 5. ОЗУ предназначено для

- a) кратковременного хранения информации в текущий момент времени;
- b) ввода и работы с информацией в текущий момент времени;
- c) длительного хранения неизменяемой информации.

### 6. Устройствами для вывода информации являются:

- a) диджитайзеры;
- b) плоттеры;
- c) мониторы.

### 7. Классификация ЭВМ по назначению:

- a) универсальные, проблемно-ориентированные, специализированные;
- b) суперЭВМ, вычислительные, гибридные;
- c) цифровые, вычислительные, универсальные.

8. В зависимости от территориального расположения абонентских систем вычислительные сети можно разделить на:
- d) глобальные, региональные, локальные;
  - e) цифровые, аналоговые, широкополосные;
  - f) локальные, дистанционные, многошинные.
9. Протокол—это
- d) набор правил, определяющий взаимодействие двух систем;
  - e) программа, определяющая взаимодействие между системами;
  - f) некоторая последовательность действий для достижения цели.
10. Языками создания гипертекстовых документов являются:
- d) HTML;
  - e) Java;
  - f) Prolog.
11. Классы программных продуктов:
- a) системное программное обеспечение, пакеты прикладных программ, инструментарий технологии программирования;
  - b) программы обслуживания сети, сервисное программное обеспечение, текстовые и графические редакторы;
  - c) базовое программное обеспечение, программные средства мультимедиа, системы искусственного интеллекта.
12. В классификации вирусов отсутствуют
- a) невидимки;
  - b) мутанты;
  - c) змеи.
13. Архивация—это
- a) помещение исходных файлов в архив;
  - b) извлечение файлов из архива;
  - c) сохранение файлов.
14. Файл—это
- a) поименованная область на диске;
  - b) часть диска;
  - c) текст или изображение.
15. Интерфейс Windows:
- a) графический;
  - b) текстовый;
  - c) оконный.
16. MS DOS—это
- a) операционная система;
  - b) оболочка операционной системы;
  - c) утилита.
17. Путь - это

- a) последовательность имен каталогов, разделенных символом " \";
  - b) последовательность имен каталогов, разделенных символом " /";
  - c) последовательность имен каталогов, разделенных символом " +".
18. Для перемещения файлов в Norton Commander необходимо нажать клавишу
- a) <F6>
  - b) <F10>
  - c) <F11>
19. В Norton Commander информация выводится в
- a) командной строке;
  - b) окна;
  - c) поля.
20. Для установления ширины полей в текстовом процессоре Word используется команда
- a) Файл;
  - b) Правка;
  - c) Вид.
21. В текстовом процессоре Word установить междустрочный интервал можно при помощи команды
- a) Формат→Шрифт;
  - b) Формат→Абзац;
  - c) Файл→Параметры страницы.
22. При копировании ячейки в Excel не меняется наименование строки. Укажите правильную запись:
- a) \$A\$5;
  - b) \$A5;
  - c) A\$5.
23. Типы данных, хранимые в ячейках Excel:
- a) текстовые, числовые, формулы, функции, даты;
  - b) только числа, даты и текст;
  - c) диаграммы, рисунки, текст, числа, даты.
24. Гипертекстовые ссылки задаются тегами:
- a) <A>...</A>
  - b) <P>...</P>
  - c) <B>...</B>
25. Каждый фрейм создается как
- a) элемент веб-страницы;
  - b) отдельная веб-страница;
  - c) специальная программа.
26. Оптимальным средством создания и хранения чертежей, схем является:
- d) векторный графический редактор;
  - e) растровый графический редактор;
  - f) фрактальный графический редактор.

## Х. ПРИМЕРНЫЕ ЭКЗАМЕНАЦИОННЫЕ БИЛЕТЫ ДЛЯ СПЕЦИАЛЬНОСТИ

040201

### АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры

Кафедра ОМиИ

« » 200 г.

Факультет ФСН

Заведующий кафедрой

Курс 1

Утверждаю: \_\_\_\_\_

Дисциплина Информатика

#### БИЛЕТ № 1

1. Информатика–предмет и задачи. Появление и развитие информатики  
Структура информатики.
2. Базовое программное обеспечение ПК. Понятия ОС- операционной системы, ОО – операционной оболочки. Базовые элементы ОС. Классификация ОС.
3. Прикладные программные продукты. Текстовый процессор. Команды Word: Параметры страницы, Форматирование абзаца, шрифта, списка, колонтитулов. Объекты (wordArt, ClipArt, Equation). Таблица. Проверка правописания.
4. Моделирование решения задач. Этапы проектирования и создания программ.
5. Найдите значение  $P$  – спроса на бытовую технику в  $n$  году в экономической модели:  $P_t = 3,4 + 5 \ln(Z) + e^{0,3t}$ , где  $Z$  –средний доход семьи в месяц.

### АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры

Кафедра ОМиИ

« » 200 г.

Факультет ФСН

Заведующий кафедрой

Курс 1

Утверждаю: \_\_\_\_\_

Дисциплина Информатика

#### БИЛЕТ № 2

1. Системы счисления: представление информации в ЭВМ.
2. Операционная система MS Dos: основные понятия, части, модули, технология работы в MS Dos.
3. Основные элементы и объекты базы данных. Типы связей.
4. Компоненты TcheckBox, TradioGroup, Tpanel, Tedit, Tlabel, Tmemo, Tbutton, TRadiogroup, TlistBox.
5. Вычислите сумму  $n$ –первых членов ряда:  $\sum_{k=1}^n \frac{1}{k^2}$ .

# XI. ПРИМЕРНЫЕ ЭКЗАМЕНАЦИОННЫХ БИЛЕТОВ ДЛЯ СПЕЦИАЛЬНОСТИ

040101

## БИЛЕТ №1

### АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры

Кафедра ОМиИ

« » 200 г.

Факультет ФСН

Заведующий кафедрой

Курс 1

Утверждаю: \_\_\_\_\_

Дисциплина Информатика

1. Информатика как наука. Этапы развития информатики.
2. Программная оболочка Norton Commander. Виды информации на панелях. Сортировка файлов.
3. ГВС, возможности, компоненты.
4. Архивация файлов. WinZip.
5. В электронной таблице

Значение формулы в ячейке D2 будет равно...

	A	B	C	D	E
1	1		=A1*5		
2		2		=A1+C3	2
3			=C1+2		

6. В Excel в ячейке C1 записана формула =\$A2-C\$2. Переместите формулу в ячейки E3, F5.

### АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Утверждено на заседании кафедры

Кафедра ОМиИ

« » 200 г.

Факультет ФСН

Заведующий кафедрой

Курс 1

Утверждаю: \_\_\_\_\_

Дисциплина Информатика

## БИЛЕТ №2

1. Направления в информатике.
2. Форматирование символов, абзацев, страниц в Word.
3. Топология ЛВС.
4. Запросы в СУБД ACCESS. Примеры.
5. В электронной таблице

Значение формулы в ячейке B2 будет равно...

	A	B	C	D	E
1	5	=A1+10	100		
2		=A1*D2+2*E3		0	2
3			=C1+2		4

6. В Excel в ячейке B2 записана формула =B\$1+\$C1. Переместите формулу в ячейки E4, D3.

**XII. КАРТА ОБЕСПЕЧЕННОСТИ ДИСЦИПЛИНЫ КАДРАМИ  
ПРОФЕССОРСКО- ПРЕПОДАВАТЕЛЬСКОГО СОСТАВА**

Ф.И.О. должность по штатному расписанию	Какое образовательное учреждение профессионального образования окончил, специальность по диплому	Ученая степень и ученое звание (почетное звание)	Стаж научно педагогической работы			Основное место работы, должность	Условия привлечения к трудовой деятельности (штатный, совместитель (внутренний или внешний с указанием доли ставки), иное)
			Всего	В т. ч. педагогический			
				Всего	В т. ч. по преподаваемой дисциплине		
3	4	5	6	7	8	9	10
Войтенко Ю.В., доцент	БТИ, инженер	к.ф-м.н., доцент	25	21	20	АмГУ доцент	Штатный
Киселева А.Н., ст.преподаватель	БГПУ, учитель математики	-	7	6	6	АмГУ, ОМИИ	Штатный