

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

НЕФТЕЮГАНСКИЙ ИНДУСТРИАЛЬНЫЙ КОЛЛЕДЖ  
(филиал) федерального государственного бюджетного образовательного  
учреждения высшего образования «Югорский государственный университет»  
(НИК (филиал) ФГБОУ ВО «ЮГУ»)


**Методические указания**


по выполнению практических работ

**ОУД.09 Информатика**

для специальности

08.02.09 «Монтаж, наладка и эксплуатация электрооборудования  
промышленных и гражданских зданий»

ОДОБРЕНО  
Предметной (цикловой)  
комиссией МиЕНД  
Протокол № 1 от 12.09 2019г.  
Председатель П(Ц)К  
  
В.В.Шумскис

УТВЕРЖДЕНО  
заседанием методсовета  
Протокол № 1 от 17.09 2019г.  
Председатель методсовета  
  
Н.И.Саватеева

Методические указания по выполнению практических работ по учебной дисциплине «Информатика» для специальности 08.02.09 «Монтаж, наладка и эксплуатация электрооборудования промышленных и гражданских зданий», разработаны в соответствии с рабочей программой учебной дисциплины «Информатика».

Организация-разработчик: Нефтеюганский индустриальный колледж (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Югорский государственный университет»

Разработчик:

Чупракова И.В. – преподаватель НИК (филиала) ФГБОУ ВО «ЮГУ»

## Содержание

Пояснительная записка .....	5
Порядок выполнения работы .....	5
Рекомендации по оформлению практической работы.....	5
Критерии оценки работ.....	5
Перечень практических работ .....	6
Практическая работа №1 Работа с образовательными информационными ресурсами .....	8
Практическая работа №2 Работа с порталом государственных услуг .....	9
Практическая работа №3 Работа с графическим интерфейсом операционной системы Windows.....	11
Практическая работа №4 Настройка операционной системы Windows. Работа с файловой структурой.....	13
Практическая работа №5 Разграничение прав доступа в сети, общее дисковое пространство в локальной сети .....	16
Практическая работа №6 Использование и настройка средств антивирусной защиты информации.....	19
Практическая работа №7 Эксплуатационные требования к компьютерному рабочему месту .....	21
Практическая работа №8 Использование систем проверки орфографии и грамматики.....	25
Практическая работа №9 Создание, организация и основные способы преобразования (верстки) текста .....	26
Практическая работа №10 Создание компьютерных публикаций с использованием готовых шаблонов .....	27
Практическая работа №11 Верстка текста с использованием процессора MS Word. Форматирование таблиц .....	29
Практическая работа №12 Создание и редактирование графических и мультимедийных объектов средствами компьютерных презентаций .....	31
Практическая работа №13 Обработка числовых данных. Применение итоговых функций .	33
Практическая работа №14 Оформление таблиц. Построение графиков функций и диаграмм различных видов .....	34
Практическая работа №15 Решение расчетных задач. Решение уравнений .....	37
Практическая работа №16 Создание базовых таблиц и межтабличных связей.....	38
Практическая работа №17 Создание запросов (на выборку, с параметром, итогового запроса).....	39
Практическая работа №18 Создание форм различными способами .....	41
Практическая работа №19 Создание отчетов .....	41
Практическая работа №20 Основные элементы языка Паскаль. Разделы описаний.....	42
Практическая работа №21 Первые программы. Ввод и вывод данных .....	46
Практическая работа №22 Оператор условия IF, оператор выбора Case .....	49
Практическая работа №23 Оператор While .....	55

Практическая работа №24 Оператор Repeat Until.....	58
Практическая работа №25 Операторы цикла For. Вложенные циклы .....	60
Практическая работа №26 Массивы. Операции с массивами. Одномерные массивы .....	64
Практическая работа №27 Массивы. Операции с массивами. Двумерные массивы.....	68
Практическая работа №28 Символьный тип. Управляющие символы .....	75
Практическая работа №29 Строковый тип. Операции над строками .....	76
Практическая работа №30 Процедуры. Функции .....	80
Практическая работа №31 Создание архива данных. Извлечение данных из архива .....	83
Практическая работа №32 Использование различных видов АСУ на практике .....	86
Практическая работа №33 Работа в браузере MS Internet Explorer.....	88
Практическая работа №34 HTML. Инструментарий. Основные понятия .....	89
Практическая работа №35 HTML. Таблицы .....	91
Практическая работа №36 HTML.Фреймы.....	93
Практическая работа №37 Осуществление поиска информации в сети Интернет.....	95
Практическая работа №38 Создание ящика электронной почты и настройка его параметров.....	97
Практическая работа №39 Использование тестирующих систем .....	99
Список литературы.....	<b>Ошибка! Закладка не определена.</b>

## **Пояснительная записка**

Методические указания по выполнению практических работ обучающимися составлены в соответствии с рабочей программой учебной дисциплины «Информатика» для специальности 08.02.09 «Монтаж, наладка и эксплуатация электрооборудования промышленных и гражданских зданий»,

Цель проведения работ – отработка необходимых навыков работы с ПК для решения прикладных задач.

### **Порядок выполнения работы**

- записать название работы, ее цель в тетрадь;
- выполнить основные задания в соответствии с ходом работы;
- выполнить дополнительные задания.

### **Рекомендации по оформлению практической работы**

- при выполнении практической работы в программе MS Word выбирать гарнитуру и размер шрифтов, выравнивание, отступы и интервалы в соответствии с заданием;
- при выполнении в программе MS Word практической работы содержащей таблицы соблюдать структуру и выравнивание ячеек таблиц, цвет границы и заливки фигур;
- при выполнении практической работы в программе MS Excel соблюдать формат и выравнивание ячеек, название листов, точность вычислений в соответствии с заданием.

Работы проводятся согласно календарно-тематическому планированию, в соответствии с учебной программой.

Пропущенные практические работы выполняются обучающимся самостоятельно и сдаются в отведенные на изучение дисциплины сроки.

При изучении теоретического материала требуется выполнение описанных операций на ПК.

### **Критерии оценки работ**

- наличие оформленной цели выполняемой работы, выполнение более половины основных заданий (удовлетворительно);
- наличие оформленной цели выполняемой работы, выполнение всех основных и более половины дополнительных заданий (хорошо);
- наличие оформленной цели выполняемой работы, выполнение всех основных и дополнительных заданий (отлично).

## Перечень практических работ

№	Тема	Наименование практических работ	Кол-во часов
1	Тема 1.1	Работа с образовательными информационными ресурсами	2
2	Тема 1.2	Работа с порталом государственных услуг	2
3	Тема 3.1	Работа с графическим интерфейсом операционной системы Windows	2
4	Тема 3.1	Настройка операционной системы Windows. Работа с файловой структурой	2
5	Тема 3.2	Разграничение прав доступа в сети, общее дисковое пространство в локальной сети	2
6	Тема 3.2	Использование и настройка средства антивирусной защиты информации	2
7	Тема 3.3	Эксплуатационные требования к компьютерному рабочему месту	2
8	Тема 4.1	Использование систем проверки орфографии и грамматики	2
9	Тема 4.1	Создание, организация и основные способы преобразования (верстки) текста	2
10	Тема 4.1	Создание компьютерных публикаций с использованием готовых шаблонов	2
11	Тема 4.1	Верстка текста с использованием процессора MS Word. Форматирование таблиц	2
12	Тема 4.2	Создание и редактирование графических и мультимедийных объектов средствами компьютерных презентаций	2
13	Тема 4.3	Обработка числовых данных. Применение итоговых функций	2
14	Тема 4.3	Оформление таблиц. Построение графиков функций и диаграмм различных видов	2
15	Тема 4.3	Решение расчетных задач. Решение уравнений	2
16	Тема 4.4	Создание базовых таблиц и межтабличных связей	2
17	Тема 4.4	Создание запросов (на выборку, с параметром, итогового запроса)	2
18	Тема 4.4	Создание форм различными способами	2
19	Тема 4.4	Создание отчетов	2
20	Тема 5.1	Основные элементы языка Паскаль. Разделы описаний	2
21	Тема 5.1	Первые программы. Ввод и вывод данных	2
22	Тема 5.1	Оператор условия IF, оператор выбора Case	2
23	Тема 5.1	Оператор While	2
24	Тема 5.1	Оператор Repeat Until	2
25	Тема 5.1	Операторы цикла For. Вложенные циклы	2
26	Тема 5.1	Массивы. Операции с массивами. Одномерные массивы	2
27	Тема 5.1	Массивы. Операции с массивами. Двумерные массивы	2
28	Тема 5.1	Символьный тип. Управляющие символы	2
29	Тема 5.1	Строковый тип. Операции над строками	2
30	Тема 5.1	Процедуры. Функции	2
31	Тема 5.1	Создание архива данных. Извлечение данных из архива	2

№	Тема	Наименование практических работ	Кол-во часов
32	Тема 5.2	Использование различных видов АСУ на практике	2
33	Тема 6.1	Работа в браузере MS Internet Explorer	2
34	Тема 6.1	HTML. Инструментарий. Основные понятия	2
35	Тема 6.1	HTML. Таблицы	2
36	Тема 6.1	HTML. Фреймы	2
37	Тема 6.2	Осуществление поиска информации в сети Интернет	2
38	Тема 6.3	Создание ящика электронной почты и настройка его параметров	2
39	Тема 6.4	Использование тестирующих систем	2
<b>Итого:</b>			<b>78</b>

## Практическая работа №1 Работа с образовательными информационными ресурсами

**Цель:** изучить образовательные информационные ресурсы

**ХОД РАБОТЫ:**

1. Изучить теоретический материал по теме «Информационные ресурсы»;
2. Составить рейтинг наиболее распространенных образовательных ресурсов, используя ресурсы сети Интернет.

### Теоретический материал

**Информационные ресурсы** это совокупность сведений, получаемых и накапливаемых в процессе развития науки и практической деятельности людей, для их многоцелевого использования в общественном производстве и управлении.

Для обеспечения своей деятельности информационные ресурсы формируют органы государственной власти, органы местного самоуправления, организации, отдельные граждане.

В информационном обществе особое внимание уделяется именно информационным ресурсам и подчеркивается их значимость по сравнению с традиционными видами ресурсов, например полезными ископаемыми. Резко повышается эффективность доступа к информационным ресурсам, становится возможным использование компьютерных телекоммуникационных сетей. По полноте предоставляемой информации к российским **информационным** ресурсам, размещенным в сети Интернет, можно отнести:

- Ресурсы органов государственной власти;
- Библиотечные ресурсы;
- Правовые ресурсы;
- Архивные ресурсы;
- Образовательные ресурсы.

Под образовательными информационными ресурсами понимают текстовую, графическую и мультимедийную информацию, а также исполняемые программы (дистрибутивы), то есть электронные ресурсы, созданные специально для использования в процессе обучения на определенной ступени образования и для определенной предметной области.

При работе с образовательными ресурсами появляются такие понятия, как субъект и объект этих ресурсов. Классификацию субъектов информационной деятельности произведем следующим образом:

- субъект, создающий объекты (все пользователи образовательной системы – преподаватель, студент);
  - субъект, использующий объекты (все пользователи образовательной системы);
  - субъект, администрирующий объекты, то есть обеспечивающий среду работы с объектами других субъектов (администраторы сети);
  - субъект, контролирующий использование объектов субъектами (инженеры).
- К образовательным электронным ресурсам можно отнести:
- учебные материалы (электронные учебники, учебные пособия, рефераты, дипломы),
  - учебно-методические материалы (электронные методики, учебные программы),
  - научно-методические (диссертации, кандидатские работы),
  - дополнительные текстовые и иллюстративные материалы (лабораторные работы, лекции),
  - системы тестирования (тесты – электронная проверка знаний),
  - электронные полнотекстовые библиотеки;



- электронные периодические издания сферы образования;
- электронные оглавления и аннотации статей периодических изданий сферы образования,
- электронные архивы выпусков.

К наиболее крупным образовательным информационным ресурсам можно отнести:

- Каталог образовательных ресурсов сети Интернет (<http://katalog.iotru>). На сайте размещена информация, с помощью которой можно получить доступ к федеральным, региональным образовательным ресурсам; проведенным и предстоящим мероприятиям, участниками которых являются педагоги и школьники, и т.д.;
- Федеральный портал «Российское образование» ([www.edu.ru](http://www.edu.ru)), содержащий каталог образовательных Интернет-ресурсов, электронный архив нормативных документов федеральных органов управления образованием, базы данных образовательных учреждений и многое другое;
- Российский общеобразовательный портал ([www.school.edu.ru](http://www.school.edu.ru)) обеспечивает оперативный доступ к образовательной информации, оказывает практическую помощь участникам образовательного процесса с применением новых образовательных технологий;
- Информационную систему «Единое окно к образовательным ресурсам» (<http://window.edu.ru>), обеспечивающую свободный доступ к интегральному каталогу образовательных интернет-ресурсов, к электронной библиотеке учебно-методических материалов для общего и профессионального образования и к ресурсам системы федеральных образовательных порталов;
- Портал «Информационно-коммуникационные технологии в образовании» ([www.ictedu.ru](http://www.ictedu.ru))? предназначенный для организации оперативного доступа к информационным ресурсам учебного, учебно-методического, справочного и информационного назначения, размещенных как на самом портале, так и на других порталах и сайтах.

## **Практическая работа №2 Работа с порталом государственных услуг**

**Цель:** изучить структуру, возможности единого портала государственных и муниципальных услуг

### **ХОД РАБОТЫ**

1. Изучить теоретический материал;
2. Заполнить таблицу 1.

#### **Теоретический материал**

Распоряжением Правительства Российской Федерации от 20 октября 2010 г. № 1815-р утверждена государственная программа Российской Федерации «Информационно-Целью программы является получение гражданами и организациями преимуществ от применения информационных и телекоммуникационных технологий за счет обеспечения равного доступа к информационным ресурсам, развития цифрового контента, применения инновационных технологий, радикального повышения эффективности государственного управления при обеспечении безопасности в информационном обществе.

**Электронное правительство** — система электронного документооборота государственного управления, основанная на автоматизации всей совокупности управленческих процессов в масштабах страны и служащая цели существенного повышения эффективности государственного управления и снижения издержек социальных коммуникаций для

каждого члена общества. Создание электронного правительства предполагает построение общегосударственной распределенной системы общественного управления, реализующей решение полного спектра задач, связанных с управлением документами и процессами их обработки.

Портал электронного правительства <http://www.gosuslugi.ru> предназначен для предоставления информации о государственных и муниципальных услугах и функциях, ведомствах, а также для оказания услуг в электронном виде (рис.1).

С помощью портала можно:

1. Получить услугу в электронном виде;
2. Получить информацию о государственной услуге, в том числе место получения, стоимость, сроки оказания и образцы документов;
3. Получить информацию о государственных и муниципальных учреждениях.

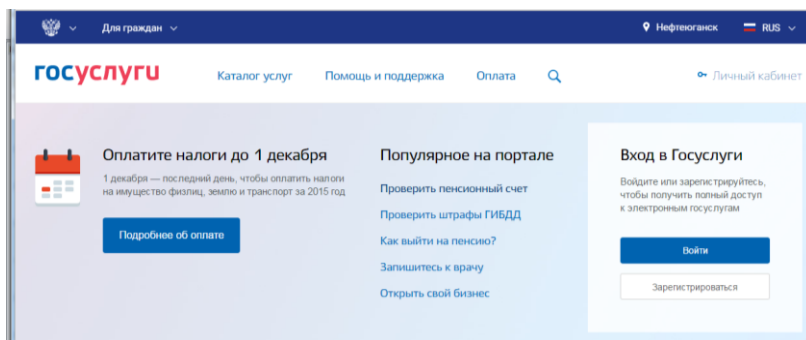


Рис. 1

Популярные госуслуги: получение загранпаспорта; оплата штрафов ГИБДД; пенсионные накопления; подача налоговой декларации; регистрация автомобиля; снятие транспортного средства с регистрации; замена паспорта РФ в 20 или 45 лет и др.

Чтобы иметь доступ ко всем услугам, необходимо зарегистрироваться на портале. Это можно сделать онлайн на <http://www.gosuslugi.ru> или в офисе ОАО «Ростелеком».

Для регистрации онлайн понадобится:

1. Проверить Страховое свидетельство государственного пенсионного страхования (СНИЛС) по базе пенсионного фонда;
2. Указать паспортные данные;
3. Подтвердить электронную почту и номер мобильного телефона;
4. Подтвердить свою личность через личное обращение, по почте России или с помощью средства электронной подписи или универсальной электронной карты.

#### **Единый портал государственных и муниципальных услуг**

Единый портал государственных и муниципальных услуг (функций) обеспечивает возможность для заявителей подавать заявления в электронной форме на получение государственных услуг, получать в электронной форме информацию о ходе рассмотрения заявлений, а также юридически значимый результат рассмотрения заявления.

Любой гражданин Российской Федерации может пройти регистрацию и получить код активации, предъявив свой паспорт и пенсионное свидетельство (СНИЛС). Дальнейшее развитие указанного порядка предполагает выдачу кодов активации государственными и муниципальными органами власти. Ресурс адаптирован для пользователей с ограниченными возможностями.

1. Загрузите портал государственных услуг Российской Федерации по адресу: <http://www.gosuslugi.ru/>
2. Откройте раздел Государственные Услуги и заполните таблицу 1:

Таблица 1

№ п/п	Наименование услуги	Способ подачи заявки	Способ получения результата	Стоимость и порядок оплаты			Сроки оказания услуги	Категории получателей	Основания для оказания услуги	Основания для отказа услуги	Результат оказания услуги	Документы, необходимые для получения услуги	
				Вид платежа	Стоимость	Вариант оплаты						Наименование документа	Количество экземпляров
1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.	Получение загранпаспорта с электронным чипом (на 10 лет)												
2.	Замена паспорта гражданина РФ												
3.	Запись на прием к врачу												
4.	Регистрация по месту жительства/пребывания												
5.	Проверка пенсионных накоплений												
6.	Получение загранпаспорта без электронного чипа (на 5 лет)												
7.	Получение и замена водительского удостоверения												

### Практическая работа №3 Работа с графическим интерфейсом операционной системы Windows

**Цель:** Изучение и работа с графическим интерфейсом операционной системы Windows

#### Теоретические сведения

##### Графический интерфейс Windows

В настоящее время все операционные системы для персональных компьютеров обеспечивают взаимодействие с пользователем с помощью графического интерфейса. Графический интерфейс позволяет осуществлять взаимодействие человека с компьютером в форме диалога с использованием окон, меню и элементов управления (диалоговых панелей, кнопок и так далее). Для работы с графическим интерфейсом используется мышь или другое координатное устройство ввода.

Основную часть экрана занимает «Рабочий стол», на котором располагаются значки и ярлыки (значки с маленькими стрелочками в нижнем левом углу). Значки и ярлыки

обеспечивают (с помощью двойного щелчка) быстрый доступ к дискам, папкам, документам, приложениям и устройствам.

## ХОД РАБОТЫ

1. Создайте на «Рабочем столе» ярлык «**Локального диска С:**», откройте объект «**Компьютер**», щелкните правой кнопкой мыши по «**Локальному диску С:**» выберите команду «**Создать ярлык**» перетащите вновь созданный ярлык на рабочий стол.
2. Аналогично создайте ярлыки часто используемых приложений и документов. перетащите их на рабочий стол.
3. Работа с программой Проводник

**Проводник** – программа ОС Windows, предназначенная для навигации по файловой структуре компьютера. Рабочая область окна Проводника имеет **панель дерева папок** (левая панель) и **панель содержимого папки** (правая панель).

Чтобы *просмотреть содержимое папки*, необходимо щелкнуть на значке папки в левой панели или дважды щелкнуть на значке папки в правой панели. Чтобы *загрузить приложение или документ*, достаточно дважды щелкнуть на значке соответствующего файла.

### Создание, удаление и переименование папок

1. На панели дерева папок выделить папку, в которой нужно создать новую;
2. Выбрать команду «**Файл – Создать – Папка**». На панели содержимого папки появится новый значок папки с текстовым полем справа (выделено прямоугольной рамкой);
3. Ввести имя папки в текстовое поле, нажать клавишу **Enter**.

### Изменить имя папки (переименовать):

1. На панели дерева папок выделить папку, имя которой нужно изменить, выбрать команду «**Файл – Переименовать**» или щелкнуть на имени папки, ввести новое имя, нажать клавишу «**Enter**».

### Удалить папку:

1. На панели дерева папок выделить удаляемую папку, выбрать команду «**Файл – Удалить**» или нажать клавишу «**Delete**», подтвердить в диалоговом окне удаление папки.

Команды переименования и удаления папки можно вызвать из контекстного меню папки.

### Выделение файлов

1. Выделение файлов выполняется только на панели содержимого папки.
2. Выделить один файл – щелкнуть на его значке.
3. Выделить несколько файлов, находящихся рядом: щелкнуть на первом по списку имени, нажать и удерживать клавишу Shift, щелкнуть на последнем по списку имени. Отменить выделение – щелкнуть вне области выделенной группы файлов.

### Создание, переименование и удаление файлов

1. Создайте файл с помощью команды «**Файл - Создать**», выберите нужный тип файла, укажите любое имя.
2. Переименуйте файл командой «**Файл – Переименовать**», введите новое имя.
3. Удалите созданный файл командой «**Файл – Удалить**» или клавишей «**Delete**».

Команды переименования и удаления файла можно вызвать из контекстного меню.

### Копирование и перенос файлов

Копирование файла – это получение копии файла в новой папке. Файлы всегда копируются из одной папки в другую.

**Перенос файла** – это перемещение файла из одной папки в другую.

1. Копирование и перенос осуществлять стандартным образом через Буфер обмена.

2. Перенос можно осуществить перетаскиванием (перемещением) выделенного файла (группы файлов) с помощью мыши.
3. Если при перетаскивании держать нажатой клавишу Ctrl, то произойдет копирование.

### Поиск файлов

**Поиск файлов** выполняется с помощью команды **Сервис/Найти/Файлы и папки...** или с помощью команды **Главное меню/Найти**.

Включение флажка **Просмотреть вложенные папки** позволит искать необходимый файл и во вложенных папках выбранной папки. Если в выпадающем списке отсутствует необходимая Вам папка, Вы можете выбрать ее вручную с помощью кнопки **Обзор...**

Ярлык

## Практическая работа №4 Настройка операционной системы Windows. Работа с файловой структурой

**Цель:** Изучить технологию настройки операционной системы и работы с файловой структурой

### ХОД РАБОТЫ

После включения компьютера и загрузки системы появляется **Рабочий стол**. На рабочем столе могут размещаться различные элементы: папки, ярлыки, значки, файлы и тд. Этот элемент операционной системы (ОС) позволяет сделать более удобным доступ, к часто используемым данным. Например, если Вы часто играете в игру или работаете с какой-то программой, то удобнее запускать ее с рабочего стола, чем каждый раз лезть в папку, где она установлена. Таким образом, на рабочем столе Вы можете разместить все часто используемые файлы для работы с ними. Но размещать эти элементы мы научимся в следующих уроках, когда сможем сами устанавливать программы и работать с папками и файлами.

#### 1. Настроить рабочий стол

- Щелкнуть правой кнопкой мыши (ПКМ) на любом месте рабочего стола выбрать команду **«Свойства»**

**В разделе «Вид»** настройте размер значков, которые располагаться на рабочем столе, так же их можно изменить размер значков, на рабочем столе нажав и удерживая клавишу “ctrl” на клавиатуре и двигая колесико мыши вверх или вниз.

- В Разделе **«Сортировка»** измените, расположение рабочего стола (начало рабочего стола – верхний левый угол экрана).
- В разделе **«Создать»** создайте на рабочем столе файл, папку или какой-то документ.
- В разделе **«Разрешение экрана»** задайте стандартное разрешение экрана. По умолчанию должны быть установлены оптимальные настройки экрана (ищите надпись «рекомендуемое» рядом со значением разрешения экрана).
- Увеличьте размер текста и других элементов экрана. Это необходимо для людей, которые плохо видят. Нажать на ссылку **«Сделать текст и другие элементы больше или меньше»** и задайте пропорцию увеличения элементов. Нажмите кнопку **«Применить»**. Система попросит Вас выйти из учетной записи и снова войти. После этого значки и текст увеличатся или уменьшатся.
- В разделе **«Персонализация»** сменить тему оформления меню рабочего стола и папок компьютера.

#### 2. «Панель задач»

- В нижней части экрана находится **«Панель задач»**. На этой панели располагается меню **«Пуск»** (левый нижний угол экрана), дата и время (правый нижний угол экрана), трей (значки левее даты и времени) и языковая панель.

- Нажав на дату или время, откроется календарь. Рядом с ним, располагаются значки некоторых программ. Обычно тут располагаются значки программ, которые запускаются при загрузке операционной системы.
  - Языковая панель нужна, для переключения языка ввода с клавиатуры. На клавиатуре расположены кнопки, на которых нарисованы буквы на русском и английском алфавитах. В зависимости от того, какой язык сейчас выбран в языковой панели, при нажатии на клавишу клавиатуры будет писать русский или английский символ. Переключать язык можно нажимая на кнопки клавиатуры. Одновременно нажать «ctrl + shift» или «alt + shift (все зависит от настроек операционной системы).
3. **Меню «Пуск»**– главное меню компьютера, тут располагаются различные программы и папки
- В левой части меню расположены самые часто используемые программы и если нажать на “Все программы” можно увидеть список всех программ, установленных на данном компьютере. Это очень удобно, ведь у Вас в одном меню собраны все программы, и их можно открыть за несколько кликов мыши.
  - Ниже списка всех программ расположено поле с поиском. введите часть названия программы (например «Блокнот»), и если она установлена, запустите ее.
  - Справа вверху находится иконка вашей учетной записи. Нажмите на текущую иконку в открывшемся окне выбрать пункт “Изменение своего рисунка”, в предложенном списке выбрать другой рисунок.
4. **Настройка меню «Пуск»**
- Для перехода к настройкам меню пуск необходимо нажать правой клавишей мыши (ПКМ) на значок меню пуск и выбрать “свойства”. Нажмите на кнопку «настройки», выберите оптимальные для своего компьютера

Контекстные меню. Объектно-ориентированный подход, используемый в операционной системе Windows, позволяет рассматривать диски, папки и файлы как объекты. Все эти объекты имеют определенные свойства, и над ними могут проводиться определенные операции.

Например, документы (документом называется любой файл, обрабатываемый с помощью приложений) имеют определенный объем и их можно копировать, перемещать и переименовывать; окна имеют размер, который можно изменять и так далее.

Хотя каждый из этих объектов имеет свои конкретные свойства и над ним возможны определенные операции, технология работы с объектами и интерфейс универсальны. Это позволяет пользователю достичь единообразия при работе с разными объектами.

Ознакомиться со свойствами объекта, а также выполнить над ним разрешенные операции можно с помощью контекстного меню. Для вызова контекстного меню необходимо осуществить правый щелчок на значке объекта.

### **Теоретический материал**

**Файл** (от англ. file – картотека, архив)– это именованная область диска для постоянного хранения информации (программ, данных для их работы, текстов, рисунков и т. д.). Каждый файл имеет имя и тип (расширение), которые записываются через точку: имя.тип.

Для каждого файла, кроме имени и расширения, ОС хранит информацию о размере файла, дате и времени его создания или последней модификации, и несколько величин, называемых атрибутами. Атрибуты – дополнительные параметры, определяющие свойства файлов: Read Only (Только для чтения); Hidden (Скрытый); System (Системный); Archive (Архивный).

Файлы по любому общему признаку, выбранному пользователем, объединяются в каталоги (папки). Каталог (папка, folder) – место на диске, в котором хранятся сведения о

файлах: их имена и их атрибуты. На логическом уровне каталоги – это элементы иерархической структуры, необходимые для обеспечения удобного доступа к файлам, особенно, если файлов на диске слишком много. Каждый каталог имеет свое имя, задаваемое пользователем при его создании. Каталог может быть вложенным, т. е. находится в каталоге более высокого уровня. Корневой каталог является самым верхним уровнем вложенности иерархической структуры и организуется на диске ОС при форматировании диска. Корневой каталог обозначается обратной косой чертой (backslash) C:\.

В отличие от имен файлов, в именах папок (директорий, каталогов) расширение обычно не ставится, так как в этом нет особой необходимости.

В пределах одной папки могут находиться сколько угодно файлов, но имена файлов вместе с расширениями должны быть уникальными, то есть не должны повторяться, но не запрещено иметь в одной папке несколько файлов с одинаковыми именами, но разными типами: письмо.txt, письмо.doc

В процессе работы с файлами возникает необходимость создавать новые файлы, заменять одни файлы другими, перемещать их с одного места на другое, переименовывать, удалять.

Windows является наиболее популярной операционной системой с графическим интерфейсом и обеспечивает возможность многозадачности - одновременной работы нескольких приложений.

## ХОД РАБОТЫ

### Управление файлами и папками

Для управления файлами и папками в Windows используют объект «Компьютер»

1. Откройте объект «Компьютер» через кнопку «Пуск». Появится окно, которое имеет стандартный внешний вид. Внутри окна располагаются объекты (рис.2). С помощью стандартного набора инструментальных средств (строка меню, лента кнопок, контекстное меню) выполняются различные операции над этими объектами, причем, как всегда, конкретная операция применяется либо ко всем, либо к выделенным объектам.
2. Самостоятельно изучите ленту кнопок «Упорядочить, Общий доступ, Запись, Новая папка, вид значков».
3. Работа с объектами:

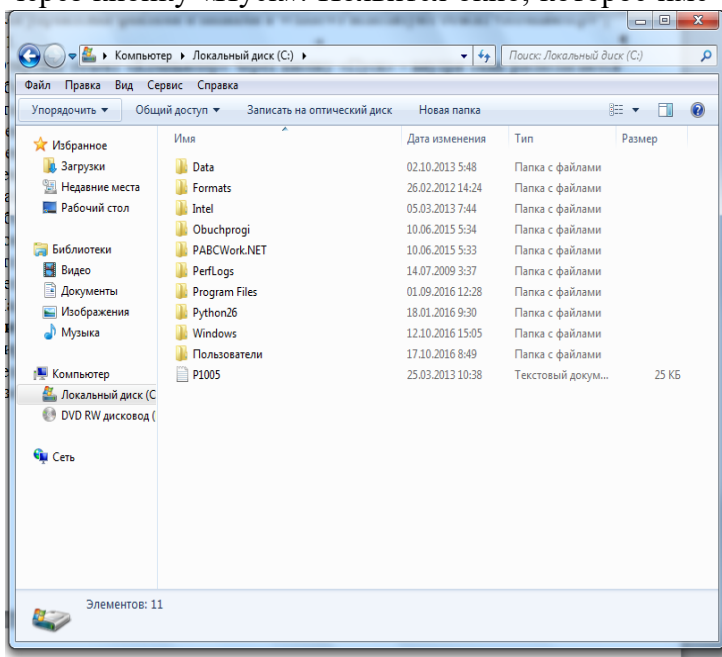


Рис. 2

#### Создание папки.

Дважды щелкните по значку Диск С (Локальный диск): - откроется окно папки Диск C:, дважды щелкните по значку папки «Документы» – откроется окно папки «Документы» на свободном месте щелкните правой кнопкой мыши. В открывшемся контекстном меню выберите команду «Создать – Папку». Появится значок папки, введите название папки – Отчет\_1. Аналогично можно создать файл: на свободном месте щелкните правой кнопкой мыши, из контекстного меню выберите команду «Создать – Документ MS Word», укажите имя документа «Карточка предприятия.doc». Также на ленте кнопок есть команда «Новая папка», которая так же

позволяет создать папку. Команда **«Файл»** в строке меню позволяет создавать и файлы и папки.

– Любым способом создайте 2 папки (Отчет 2015\_2, Отчет 2015\_3) и 2 файла (Список сотрудников.xls, Типовой договор.doc).

#### **Переименование файлов и папок**

– Щелкните правой кнопкой мыши на значке Отчет-1. в открывшемся контекстном меню выберите пункт **«Переименовать»**. Дайте папке имя Отчет\_2015.

– Выделите значок Отчет\_2015 в строке меню выберите команду **«Файл - Переименовать»** укажите папке имя Отчет\_2015\_1.

#### **Перемещение файлов и папок**

– Выделите папку Отчет\_2015\_1 в окне папки **«Документы»**. В строке меню выберите команду **«Правка- Вырезать»**. (Таким способом объект помещается в буфер обмена с удалением из текущей папки).

– Откройте **«Локальный диск С:»** и в строке меню выберите команду **«Правка – Вставить»** (объект будет вставлен из буфера обмена).

– Команды **«Вырезать (Ctrl+X) – Вставить (Ctrl+C)»** также доступны при вызове контекстного меню.

#### **Копирование файлов и папок**

– Выделите файл **«Карточка предприятия.doc»** строке меню выберите команду **«Правка – Копировать»**. (Таким способом объект помещается в буфер обмена без удаления). Откройте папку Отчет\_2015\_1, в ней в строке меню выберите команду **«Правка – Вставить»**. (Будет вставлен объект из буфера обмена).

– Скопируйте файл: Список сотрудников.xls, Типовой договор.doc в папки Отчет 2015\_2, Отчет 2015\_3

#### **Удаление файлов и папок**

– В окне папки Отчет 2015\_3 удалите файл - **«Карточка предприятия.doc»**. Для этого щелкните правой кнопкой мыши на значке папки и в открывшемся контекстном меню выберите пункт **«Удалить»**. В открывшемся диалоговом окне подтвердите необходимость удаления объекта.

– Удалите все созданные вами папки, и файлы.

## **Практическая работа №5 Разграничение прав доступа в сети, общее дисковое пространство в локальной сети**

**Цель:** Изучение принципов взаимодействия компьютеров в сети; выполнить настройку общего доступа к принтеру локальной сети

### **Теоретический материал**

Существуют две основные цели использования локальных сетей:

1. Обмен файлами между пользователями сети;
2. Использование общих ресурсов, доступных всем пользователям сети: большого пространства дисковой памяти, принтеров, централизованной базы данных, программного обеспечения и других.

Пользователей локальной сети принято называть рабочей группой, а компьютеры, за которыми они работают — **«рабочими станциями»** или **«клиентами»**.

Если все компьютеры в сети равноправны, то есть сеть состоит только из рабочих станций пользователей, то ее называют **одноранговой сетью**. **Одноранговые сети** используются для реализации первой из отмеченных целей — обмена файлами. У каждого компьютера в такой сети есть свое имя. Члены рабочей группы могут обращаться по этим



именам к дисковой памяти ПК своих коллег и копировать файлы на свой компьютер или копировать свои файлы на другие компьютеры. Таким образом, одноранговая локальная сеть избавляет от необходимости использовать дискеты для переноса информации с одного компьютера на другой. Такой режим особенно удобен при совместной работе членов рабочей группы над общим проектом.

Работу одноранговой сети поддерживают операционные системы Windows выбрав режим «Сетевое окружение», пользователь получает на экране меню со списком имен подключенных к сети компьютеров. Далее он может обращаться к дискам любого из них, работать с размещенными на них файлами как с файлами собственного ПК. Кроме того, через сеть можно получить доступ к внешним устройствам, подключенным к другим ПК, например, к принтеру.

Другой способ организации локальной сети — **сеть с выделенным** (главным) компьютером.

Его называют **сервером**. Как и в глобальных сетях, слово «сервер» обозначает компьютер (и его программное обеспечение), предоставляющий определенные услуги пользователям сети. Услуги могут быть самыми разными: хранение и рассылка файлов, доступ к централизованной базе данных, вычислительные услуги и пр.

Обычно сервер — это более мощная машина, чем рабочие станции, с большим жестким диском, дополнительными внешними устройствами (например, дисководом для компакт-дисков, принтером, модемом). При такой организации локальной сети реализуется вторая из отмеченных выше целей: доступ пользователей к общим аппаратным и информационным ресурсам сервера. В частности, программы, хранящиеся на диске сервера, могут загружаться в оперативную память рабочей станции и запускаться на выполнение подобно тому, как это делается с собственного диска ПК. Находясь на своем рабочем месте, пользователь может создавать файлы и сохранять их на жестком диске сервера, загружать файлы в свой компьютер, использовать сетевой принтер и так далее.

Работой локальной сети управляет сетевая операционная система, которая поддерживает стандарты (протоколы) обмена информацией в сети, устанавливает очередность при обращении различных пользователей к одним и тем же ресурсам, защищает ресурсы от несанкционированного доступа и пр.

При выборе сервера следует обращать внимание на множество показателей. Если вы возьмете большой корпус, быстрые контроллеры и диски, это отнюдь не сделает компьютер сервером. Серверы изначально проектируются, собираются и тестируются с требованием максимальной пропускной способности, надежности, отказоустойчивости и расширяемости. Соответственно их цена обычно в 1,5-3 раза выше, чем цена стандартных компьютеров сравнимой конфигурации. Не гонитесь за дешевизной!

Для крупных организаций, основой работы которых является база данных, находящаяся на сервере локальной сети, остановка его работы даже на час может привести к большим финансовым потерям. Поэтому стоит внимательно присмотреться к средствам дублирования серверов как целого. Наиболее дешевым является наличие резервного компьютера (более слабого) с предустановленной идентичной операционной системой. В случае сбоя основного сервера подключается резервный, информация восстанавливается средствами резервного копирования, и фирма продолжает работать. Недостатком в данном случае является то, что обычно имеется вчерашняя копия и будут потеряны все сегодняшние изменения. Более практичным является сочетание дублирования дисковой подсистемы (т. е. на дисках информация сохранится в любом случае) и идентичных дисковых подсистем в основном и резервном серверах — вы сможете просто переставить диски с основного сервера и включить резервный.

**Разграничением** (логическим управлением) **доступа** называется процесс установок полномочий (совокупности прав) субъекта для последующего контроля санкционированного использования вычислительных ресурсов, доступных в автоматизированной системе. Обычно полномочия субъекта представляются: списком ресурсов, доступных поль-

зователю, и правами по доступу к каждому ресурсу из списка. В качестве вычислительных ресурсов могут быть программы, информация, логические устройства, объем памяти, время процессора, приоритет и т. д.

Обычно выделяют следующие методы разграничения доступа:

1. Разграничение доступа по спискам;
2. Использование матрицы установления полномочий;
3. По уровням секретности и категориям;
4. Парольное разграничение доступа.

При разграничении доступа по спискам задаются соответствия:

1. Каждому пользователю – список ресурсов и прав доступа к ним;
  2. Каждому ресурсу – список пользователей и их прав доступа к данному ресурсу.
- Использование матрицы установления полномочий подразумевает применение матрицы доступа (таблицы полномочий). Данный метод предоставляет более унифицированный и удобный подход, т. к. вся информация о полномочиях хранится в виде единой таблицы, а не в виде разнотипных списков. Недостатками матрицы являются ее возможная громоздкость и не совсем оптимальное использование ресурсов (большинство клеток – пустые).

Разграничения доступа по уровням секретности и категориям состоят в том, что ресурсы АС разделяются в соответствии с уровнями секретности или категорий. При разграничении по уровню секретности выделяют несколько уровней, например: общий доступ, конфиденциально, секретно, совершенно секретно. При разграничении по категориям задается и контролируется ранг категории, соответствующей пользователю. Соответственно, все ресурсы АС декомпозируют по уровню важности, причем определенному уровню соответствует некоторый ранг персонала (типа: руководитель, администратор, пользователь).

Парольное разграничение, очевидно, представляет использование методов доступа субъектов к объектам по паролю. При этом используются все методы парольной защиты. Очевидно, что постоянное использование паролей создает неудобства пользователям и временные задержки. Поэтому указанные методы используют в исключительных ситуациях.

На практике обычно сочетают различные методы разграничения доступа. Например, первые три метода усиливают парольной защитой.

## ХОД РАБОТЫ

1. Определите общий ресурс компьютера.
  - В операционной системе Windows найдите на рабочем столе значок Сеть или в главном меню «Пуск».
  - Открыть папку, где будут видны все компьютеры, которые подключены в одну сеть.
  - В данном окне появятся все компьютеры, которые подключены к сети.
  - Открыть один из них. Посмотреть ресурсы компьютера, которыми можно воспользоваться. Такие ресурсы называются общими.
2. Предоставьте доступ для пользователей локальной сети к папке на своем компьютере, подключенном к локальной сети:
  - В операционной системе Windows открыть окно папки Компьютер и на одном из дисков С: или D: создать свою папку. Назвать ее номером своей группы.
  - Щелкнуть правой кнопкой мыши по значку папки и в контекстном меню папки выбрать команду Общий доступ.
  - В появившемся диалоговом окне Дополнительный общий доступ установить флажок Открыть общий доступ к этой папке.

- Если все правильно сделано, то на диске (у вашей папки) появится значок, который показывает, что папка является общей.
3. Проверьте возможности доступа к ресурсам компьютеров, подключенных к локальной сети:
    - Щелкнуть по значку Сеть, в окне появится список компьютеров, подключенных к локальной сети.
    - Открыть свой компьютер и внимательно посмотреть: какие из ресурсов доступны пользователям. Если название Вашей папки есть в перечне, то все сделано правильно.
  4. Заполните таблицу

Указать основное назначение компьютерной сети.	
Указать основную характеристику каналов связи.	
Указать объект, который является абонентом сети.	

## **Практическая работа №6 Использование и настройка средств антивирусной защиты информации**

**Цель:** Изучение средств антивирусной защиты информации

### **Теоретический материал**

Вредоносная программа — компьютерная программа или переносной код, предназначенный для реализации угроз информации, хранящейся в компьютерной системе, либо для скрытого нецелевого использования ресурсов системы, либо иного воздействия, препятствующего нормальному функционированию компьютерной системы. К вредоносному программному обеспечению относятся сетевые черви, классические файловые вирусы, троянские программы, хакерские утилиты и прочие программы, наносящие вред компьютеру, на котором они запускаются на выполнение, или другим компьютерам в сети.

Независимо от типа, вредоносные программы способны наносить значительный ущерб, реализуя любые угрозы информации — угрозы нарушения целостности, конфиденциальности, доступности.

1. **Классические компьютерные вирусы.** К данной категории относятся программы, распространяющие свои копии по ресурсам локального компьютера с целью:
  - последующего запуска своего кода при каких-либо действиях пользователя;
  - дальнейшего внедрения в другие ресурсы компьютера.
2. **Сетевые черви.** К данной категории относятся программы, распространяющие свои копии по локальным и/или глобальным сетям с целью:
  - проникновения на удаленные компьютеры;
  - запуска своей копии на удаленном компьютере;
  - дальнейшего распространения на другие компьютеры в сети.

Для своего распространения сетевые черви используют разнообразные компьютерные и мобильные сети: электронную почту, системы обмена мгновенными сообщениями, файлообменные (P2P) и IRC-сети, LAN, сети обмена данными между мобильными устройствами (телефонами, карманными компьютерами) и т. д.

3. **Троянские программы.** В данную категорию входят программы, осуществляющие различные несанкционированные пользователем действия: сбор информации и ее передачу злоумышленнику, ее разрушение или злонамеренную модификацию, нарушение работоспособности компьютера, использование ресурсов компьютера в неблагоприятных целях.
4. **Хакерские утилиты** и прочие вредоносные программы. К данной категории относятся:
  - утилиты автоматизации создания вирусов, червей и троянских программ (конструкторы);
  - программные библиотеки, разработанные для создания вредоносного ПО;
  - хакерские утилиты скрытия кода зараженных файлов от антивирусной проверки (шифровальщики файлов);
  - «злые шутки», затрудняющие работу с компьютером;
  - программы, сообщающие пользователю заведомо ложную информацию о своих действиях в системе;
  - прочие программы, тем или иным способом намеренно наносящие прямой или косвенный ущерб данному или удалённым компьютерам.

Руткит (Rootkit) - программа или набор программ, использующих технологии сокрытия системных объектов (файлов, процессов, драйверов, сервисов, ключей реестра, открытых портов, соединений и пр.) посредством обхода механизмов системы.

В системе Windows под термином руткит принято считать программу, которая внедряется в систему и перехватывает системные функции, или производит замену системных библиотек. Перехват и модификация низкоуровневых API функций в первую очередь позволяет такой программе достаточно качественно маскировать свое присутствие в системе, защищая ее от обнаружения пользователем и антивирусным ПО. Кроме того, многие руткиты могут маскировать присутствие в системе любых описанных в его конфигурации процессов, папок и файлов на диске, ключей в реестре. Многие руткиты устанавливают в систему свои драйверы и сервисы (они естественно также являются «невидимыми»).

Современные антивирусные программы обеспечивают комплексную защиту программ и данных на компьютере от всех типов вредоносных программ и методов их проникновения на компьютер (Интернет, локальная сеть, электронная почта, съемные носители информации). Большинство антивирусных программ сочетает в себе функции постоянной защиты (антивирусный монитор) и функции защиты по требованию пользователя (антивирусный сканер).

Межсетевой экран — это программа, установленная на пользовательском компьютере и предназначенная для защиты от несанкционированного доступа к компьютеру. Другое распространенное название сетевого экрана — файервол от английского термина firewall. Иногда сетевой экран называют еще брандмауэром (нем. brandmauer) — это немецкий эквивалент слова firewall. Основная задача сетевого экрана — не пропускать (фильтровать) пакеты, не подходящие под критерии, определённые в конфигурации сетевого экрана. Межсетевой экран позволяет:

- Блокировать хакерские атаки;
- Не допускать проникновение сетевых червей;
- Препятствовать троянским программам отправлять конфиденциальную информацию о пользователе и компьютере.

В операционной системе Windows проверить выбранные объекты на наличие вредоносных объектов, выполнить лечение или удаление зараженных объектов

- Запустить на выполнение антивирусную программу.
- Запустить обновление из контекстного меню.
- Выполнить проверку съемного носителя.
- Выполнить проверку локального диска.
- Отчет о работе антивирусной содержит информацию о результатах проверки.

## **Практическая работа №7 Эксплуатационные требования к компьютерному рабочему месту**

**Цель:** ознакомиться с эксплуатационными требованиями к компьютерному рабочему месту

### **Теоретический материал**

На рабочих местах пользователей персональных компьютеров должны обеспечиваться оптимальные параметры микроклимата в соответствии с СанПин 2.2.4.548-96. Согласно этому документу для категории тяжести работ 1а температура воздуха должна быть в холодный период года не более 22-24оС, в теплый период года 20-25оС. Относительная влажность должна составлять 40-60%, скорость движения воздуха - 0,1 м/с. Для поддержания оптимальных значений микроклимата используется система отопления и кондиционирования воздуха. Для повышения влажности воздуха в помещении следует применять увлажнители воздуха или емкости с питьевой водой.

На рабочих пользователей должно быть естественное и искусственное освещение. Световой поток из оконного проема должен падать на рабочее место оператора с левой стороны.

Искусственное освещение в помещениях эксплуатации компьютеров должно осуществляться системой общего равномерного освещения.

Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана.

Отраженная блескость на рабочих поверхностях ограничивается за счет правильного выбора светильника и расположения рабочих мест по отношению к естественному источнику света.

Для искусственного освещения помещений с персональными компьютерами следует применять светильники типа ЛПО36 с зеркализированными решетками, укомплектованные высокочастотными пускорегулирующими аппаратами. Допускается применять светильники прямого света, преимущественно отраженного света типа ЛПО13, ЛПО5, ЛСО4, ЛПО34, ЛПО31 с люминисцентными лампами типа ЛБ. Допускается применение светильников местного освещения с лампами накаливания. Светильники должны располагаться в виде сплошных или прерывистых линий сбоку от рабочих мест параллельно линии зрения пользователя при разном расположении компьютеров.

Для обеспечения нормативных значений освещенности в помещениях следует проводить чистку стекол оконных проемов и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

Уровни шума на рабочих местах пользователей персональных компьютеров не должны превышать значений, установленных СанПиН 2.2.4/2.1.8.562-96 и составляют не более 50 дБА. Снизить уровень шума в помещениях можно использованием звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот 63-8000 Гц для отделки стен и потолка помещений. Дополнительный звукопоглоща-

ющий эффект создают однотонные занавески из плотной ткани, повешенные в складку на расстоянии 15-20 см от ограждения. Ширина занавески должна быть в 2 раза больше ширины окна.

Рабочие места с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, желательно слева.

Схемы размещения рабочих мест с персональными компьютерами должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м.

Рабочий стол может быть любой конструкции, отвечающей современным требованиям эргономики и позволяющей удобно разместить на рабочей поверхности оборудование с учетом его количества, размеров и характера выполняемой работы. Целесообразно применение столов, имеющих отдельную от основной столешницы специальную рабочую поверхность для размещения клавиатуры. Используются рабочие столы с регулируемой и нерегулируемой высотой рабочей поверхности. При отсутствии регулировки высота стола должна быть в пределах от 680 до 800 мм.

Глубина рабочей поверхности стола должна составлять 800 мм (допускаемая не менее 600 мм), ширина - соответственно 1 600 мм и 1 200 мм. Рабочая поверхность стола не должна иметь острых углов и краев, иметь матовую или полуматовую фактуру.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной - не менее 500 мм, глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног - не менее 650 мм.

Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 градусов вниз от горизонтали).

Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю.

Для удобства считывания информации с документов применяются подвижные подставки (пюпитры), размеры которых по длине и ширине соответствуют размерам устанавливаемых на них документов. Пюпитр размещается в одной плоскости и на одной высоте с экраном.

Для обеспечения физиологически рациональной рабочей позы, создания условий для ее изменения в течение рабочего дня применяются подъемно-поворотные рабочие стулья с сиденьем и спинкой, регулируемые по высоте и углам наклона, а также расстоянию спинки от переднего края сидения.

Рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 град. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Количество и длительность регламентированных перерывов, их распределение в течение рабочей смены устанавливается в зависимости от категории работ на ПК и продолжительности рабочей смены.

При 8-часовой рабочей смене и работе на ПК регламентированные перерывы следует устанавливать:

- для первой категории работ через 2 часа от начала смены и через 2 часа после обеденного перерыва продолжительностью 15 минут каждый;
- для второй категории работ - через 2 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва продолжительностью 15 минут каждый или продолжительностью 10 минут через каждый час работы;

– для третьей категории работ - через 1,5- 2,0 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

При 12-часовой рабочей смене регламентированные перерывы должны устанавливаться в первые 8 часов работы аналогично перерывам при 8-часовой рабочей смене, а в течение последних 4 часов работы, независимо от категории и вида работ, каждый час продолжительностью 15 минут.

Продолжительность непрерывной работы на ПК без регламентированного перерыва не должна превышать 2 часа.

При работе на ПК в ночную смену продолжительность регламентированных перерывов увеличивается на 60 минут независимо от категории и вида трудовой деятельности.

Эффективными являются нерегламентированные перерывы (микропаузы) длительностью 1-3 минуты.

Регламентированные перерывы и микропаузы целесообразно использовать для выполнения комплекса упражнений и гимнастики для глаз, пальцев рук, а также массажа. Комплексы упражнений целесообразно менять через 2-3 недели.

Пользователям ПК, выполняющим работу с высоким уровнем напряженности, показана психологическая разгрузка во время регламентированных перерывов и в конце рабочего дня в специально оборудованных помещениях (комнатах психологической разгрузки).

Медико-профилактические и оздоровительные мероприятия.

Все профессиональные пользователи ПК должны проходить обязательные предварительные медицинские осмотры при поступлении на работу, периодические медицинские осмотры с обязательным участием терапевта, невропатолога и окулиста, а также проведением общего анализа крови и ЭКГ.

Не допускаются к работе на ПК женщины со времени установления беременности и в период кормления грудью.

Близорукость, дальнозоркость и другие нарушения рефракции должны быть полностью скорректированы очками. Для работы должны использоваться очки, подобранные с учетом рабочего расстояния от глаз до экрана дисплея. При более серьезных нарушениях состояния зрения вопрос о возможности работы на ПК решается врачом-офтальмологом.

Для снятия усталости аккомодационных мышц и их тренировки используются компьютерные программы типа Relax.

Интенсивно работающим целесообразно использовать такие новейшие средства профилактики зрения, как очки ЛПО-тренер и офтальмологические тренажеры ДАК и «Снайпер-ультра».

Досуг рекомендуется использовать для пассивного и активного отдыха (занятия на тренажерах, плавание, езда на велосипеде, бег, игра в теннис, футбол, лыжи, аэробика, прогулки по парку, лесу, экскурсии, прослушивание музыки и т.п.). Дважды в год (весной и поздней осенью) рекомендуется проводить курс витаминотерапии в течение месяца. Следует отказаться от курения. Категорически должно быть запрещено курение на рабочих местах и в помещениях с ПК.

Обеспечение электробезопасности и пожарной безопасности на рабочем месте

На рабочем месте пользователя размещены дисплей, клавиатура и системный блок. При включении дисплея на электронно-лучевой трубке создается высокое напряжение в несколько киловольт. Поэтому запрещается прикасаться к тыльной стороне дисплея, вытирать пыль с компьютера при его включенном состоянии, работать на компьютере во влажной одежде и влажными руками.

Перед началом работы следует убедиться в отсутствии свешивающихся со стола или висящих под столом проводов электропитания, в целостности вилки и провода электропитания, в отсутствии видимых повреждений аппаратуры и рабочей мебели.

Токи статического электричества, наведенные в процессе работы компьютера на корпусах монитора, системного блока и клавиатуры, могут приводить к разрядам при прикосновении к этим элементам. Такие разряды опасности для человека не представляют, но могут привести к выходу из строя компьютера. Для снижения величин токов статического электричества используются нейтрализаторы, местное и общее увлажнение воздуха, использование покрытия полов с антистатической пропиткой.

Пожарная безопасность - состояние объекта, при котором исключается возможность пожара, а в случае его возникновения предотвращается воздействие на людей опасных его факторов и обеспечивается защита материальных ценностей.

Противопожарная защита - это комплекс организационных и технических мероприятий, направленных на обеспечение безопасности людей, предотвращение пожара, ограничение его распространения, а также на создание условий для успешного тушения пожара.

Пожарная безопасность обеспечивается системой предотвращения пожара и системой пожарной защиты. Во всех служебных помещениях обязательно должен быть «План эвакуации людей при пожаре», регламентирующий действия персонала в случае возникновения очага возгорания и указывающий места расположения пожарной техники.

Пожары в вычислительных центрах (ВЦ) представляют особую опасность, так как сопряжены с большими материальными потерями. Характерная особенность

ВЦ - небольшие площади помещений. Как известно, пожар может возникнуть при взаимодействии горючих веществ, окислителя и источников зажигания. В помещениях ВЦ присутствуют все три основных фактора, необходимые для возникновения пожара.

Горючими компонентами на ВЦ являются: строительные материалы для акустической и эстетической отделки помещений, перегородки, двери, полы, изоляция кабелей и др.

Источниками зажигания в ВЦ могут быть электрические схемы от ЭВМ, приборы, применяемые для технического обслуживания, устройства электропитания, кондиционирования воздуха, где в результате различных нарушений образуются перегретые элементы, электрические искры и дуги, способные вызвать загорания горючих материалов.

В современных ЭВМ очень высокая плотность размещения элементов электронных схем. В непосредственной близости друг от друга располагаются соединительные провода, кабели. При протекании по ним электрического тока выделяется значительное количество теплоты. При этом возможно оплавление изоляции. Для отвода избыточной теплоты от ЭВМ служат системы вентиляции и кондиционирования воздуха. При постоянном действии эти системы представляют собой дополнительную пожарную опасность.

Для большинства помещений ВЦ установлена категория пожарной опасности В.

Одна из наиболее важных задач пожарной защиты - защита строительных помещений от разрушений и обеспечение их достаточной прочности в условиях воздействия высоких температур при пожаре. Учитывая высокую стоимость электронного оборудования ВЦ, а также категорию его пожарной опасности, здания для ВЦ и части здания другого назначения, в которых предусмотрено размещение ЭВМ, должны быть первой и второй степени огнестойкости. Для изготовления строительных конструкций используются, как правило, кирпич, железобетон, стекло, металл и другие негорючие материалы. Применение дерева должно быть ограничено, а в случае использования необходимо пропитывать его огнезащитными составами.



## ХОД РАБОТЫ

Заполните таблицу

Перечислите основные санитарно-гигиенические требования к рабочему месту пользователя (не менее 6)	
Укажите некоторые требования к помещениям в которых работают пользователи на ПК	
Укажите, какие действия запрещены при работе с ПК	

### Практическая работа №8 Использование систем проверки орфографии и грамматики

**Цель:** Изучение систем проверки орфографии и грамматики

#### Теоретический материал

Системы проверки правописания, используемые в большинстве современных текстовых редакторов, позволяют выявлять значительную часть допущенных пользователем опечаток и орфографических ошибок. Принцип действия типичной системы проверки правописания следующий: во встроенном словаре системы содержится большой набор слов анализируемого языка в различных грамматических формах (время, число и т.д.), система пытается найти проверяемое слово в этом словаре. Если слово найдено, то оно считается правильно написанным. Если слово не найдено в словаре, но есть похожие слова, то выдается сообщение об ошибке и предлагаются возможные варианты замены. Если ничего похожего не найдено, то система предлагает исправить слово или занести его в словарь. Конечно, принцип автоматизированной проверки орфографии здесь изложен весьма в упрощенной форме, но суть его именно такова. При проверке правописания слова с ошибками были подчеркнуты красной линией.

Метод проверки орфографии по словарю не позволяет выявить случаи, когда ошибка допущена таким образом, что получившееся слово есть в словаре. Часто такие ошибки легко заметны человеку, но совершенно скрыты от программы.

Существуют более интеллектуальные системы проверки правописания, позволяющие выявлять ошибки в согласовании форм слов и расстановке знаков препинания (то есть в грамматике и пунктуации). В них хранится набор соответствующих правил, записанных в формальном виде. Такая система смогла бы обратить внимание пользователя на подозрительное место в фразах или пропущено связующее слово между ним и предложением). В любом случае, пользователь обратит внимание на эту фразу и исправит.

## ХОД РАБОТЫ

1. Запустите программу MS Word, напечатайте текст, не обращая внимания на ошибки и несоответствия.

Вредоносная программа — компьютерная программа или переносной код, предназначенный для реализации угроз информации, хранящейся в компьютерной системе, либо для скрытого нецелевого использования ресурсов системы, либо иного воздействия, препятствующего нормальному функционированию компьютерной системы. К вредоносному программному обеспечению относятся сетевые черви, классические файловые вирусы, троянские программы, хакерские утилиты и прочие программы, наносящие вред компьютеру, на котором они запускаются на выполнение, или другим компьютерам в сети. Независимо от типа, вредоносные программы способны наносить значительный ущерб, реализуя любые угрозы информации — угрозы нарушения целостности, конфиденциальности, доступности.

2. Обратите внимание, что некоторые слова в тексте подчеркнуты красной волнистой линией или зеленой волнистой линией. Значит Ваш Word настроен на автоматическую проверку орфографии и грамматики. Красная линия – орфографическая ошибка. Щелкните правой кнопкой по подчеркнутому слову. Встроенный словарь предлагает слова для замены, вы выбираете слово из списка, если же слово набрано, верно, и в этом уверены, можете его «Добавить в словарь». Если сомневаетесь, нажмите «Пропустить все».

### **Практическая работа №9 Создание, организация и основные способы преобразования (верстки) текста**

**Цель:** Изучение технологии создания, организации и преобразования текста

## ХОД РАБОТЫ

1. Создать новый документ:
  - Запустить программу: выбрать меню «**Пуск**» команда «**Все программы**»-«**MS Office**»-«**MS Word**»;
  - При запуске программы MS Word новый документ создается автоматически, для создания еще одного нового документа необходимо в меню «**Файл**» выбрать команду «**Создать**»-«**Новый документ**» нажать кнопку «**Создать**».

2. **Сохранение документа:**

- Для создания документа необходимо в меню «**Файл**» выбрать команду «**Сохранить**» в появившемся диалоговом окне указать имя файла и папку где будет храниться документ.

3. **Использование табуляции**

Установка позиций табуляции:

- Выделите абзацы, в которых необходимо установить позиции табуляции.
- Щелкните горизонтальную линейку в том месте, где необходимо установить позицию табуляции.

Изменение позиций табуляции:

- Выберите в меню *Формат* команду *Табуляция*.
- В появившемся диалоговом окне *Табуляция* Вы можете изменить размеры ранее созданных позиций табуляции, создать новые, удалить ненужные позиции, выбрать тип выравнивания для позиций табуляции.

**Выполните задание по образцу, используя таблицу:**

Наименование	Цена	Количество
Ручка шариковая	5 р.	148
Карандаш простой	3 р.	154
Карандаш клеящий	7 р.	45
Ножницы	15 р.	63

#### 4. Текстовые эффекты

Оформление текста:

- Выделите текстовый фрагмент.
- В меню *Формат* выберите команду *Шрифт*.
- В открывшемся диалоговом окне *Шрифт* на вкладке *Шрифт* выберите нужный вид подчеркивания из списка *Подчеркивание*.
- Выберите нужные эффекты с помощью группы флажков *Эффекты*.
- Выберите нужный интервал между символами, используя вкладку *Интервал* диалогового окна *Шрифт*.
- Щелкните кнопку **ОК**.

**Наберите текст, соблюдая оформление (шрифт: Таhoma, 12):**

**Автоматизированное рабочее место (АРМ)** преподавателя/учащегося – совокупность средств вычислительной техники и *программно-аппаратных* средств, предназначенная для работы **одного** человека.

**Аппаратное обеспечение** – материальное оборудование, из которого состоит ЭВМ (процессор, клавиатура, дисковод, печатающее устройство и т.д.)

#### 5. Оформление и заливка:

- Выделите фрагмент текста или таблицы.
- В меню *Формат* выберите команду *Границы и заливка*.
- В появившемся диалоговом окне *Границы и заливка* на вкладке *Граница* измените тип, цвет и ширину линий оформления.
- На вкладке *Заливка* выберите тип заливки и щелкните кнопку **ОК**.

**Наберите текст и добавьте оформление и заливку (шрифт: Arial, 10):**

**База данных (БД)** – именованная совокупность данных, которая отображает состояние объектов и их отношений в данной предметной области. БД обеспечивает использование одних и тех же данных в различных приложениях, допускает решение задач планирования, проектирования, исследования, управления.

**База знаний (БЗ)** – организованная *совокупность знаний*, представленная в форме, которая допускает *автоматическое* или *автоматизированное* использование этих знаний на основе реализации возможностей средств **информационных технологий**.

### Практическая работа №10 Создание компьютерных публикаций с использованием готовых шаблонов

**Цель:** Научиться создавать компьютерные публикации на основе готовых шаблонов

## ХОД РАБОТЫ


1. Создать документ на основе существующего шаблона через меню «Файл» - «Создать» - «Образцы шаблонов» - «Современный факс»

Нефтеюганск, ул. Строителей №1	Нефтеюганский Индустриальный Колледж			
<b>Факсимильное сообщение</b>				
Кому: Нестеровой П.В.	от: Иванова И.Н.			
Факс: 456789	Дата: 10. Января 2004 г.			
Тел.: [Введите № телефона]	Стр.: [Введите число страниц]			
На: [Введите тему сообщения]	Копия: [Введите имя]			
Срочно	Секретно	На утверждение	Требует ответа	Разослать

2. Создать шаблон Служебной записки (из наиболее похожего шаблона). Сохранить в своей папке.

Нефтеюганский Индустриальный колледж
<b>Служебная записка</b>
<ul style="list-style-type: none"><li>• Кому: гл. инженеру Друзь И.Н.</li><li>• от: мастера п/о Иванова И.И.</li><li>• Копия: зам директору Сиротиной Т.А</li><li>• Дата: 15.11.03</li><li>• На: О покупке технических средств в кабинет В.Т.</li></ul>
Прошу преобрести в кабинет ВТ следующие технические средства: принтер лазерный, сканер, ПЗВМ 486-15 штук.





3. На основе текстового документа создать шаблон Заявления руководителю вашей организации.
4. Создать шаблон с собственным Резюме (используя готовый шаблон).
5. Придумать и создать шаблон Фирменного бланка учреждения, используя образец, изображенный на рисунке:

 <p>Министерство общего и профессионального образования РФ Главное управление образования администрации Ханты-Мансийского автономного округа Сургутский профессиональный колледж</p>	
«    »                    200 г.                    №	
Россия, 626400, Тюменская обл., г. Сургут, ул. Маяковского, д. 41	t/fax (3462) 33-83-26 директор t/fax (3462) 33-87-84 бухгалтерия p/c 14131419 к/с 2142423 БИК 047144000 в РКЦ г. Сургута ИНН 8602200072 код по ОКОНХ 92200 код по ОКПО 03288094

## Практическая работа №11 Верстка текста с использованием процессора MS Word. Форматирование таблиц

**Цель:** Изучить технологию верстки текста и форматирования таблиц

### ХОД РАБОТЫ

1. Введите с клавиатуры заданный текст (см. на следующей странице), не обращая пока внимания на выравнивание, маркированные списки и свойства шрифтов.
2. Создайте 4 своих стиля:
  - **МойСтиль1**, основанный на стиле «Заголовок 1», в котором задайте:
    - шрифт: Times New Roman, 16; полужирный;
    - абзац: выравнивание по центру; отступы слева и справа – 0 см; первая строка – нет; интервал перед и после – 6 пт, межстрочный – одинарный.
  - **МойСтиль2**, основанный на стиле «Заголовок 2», в котором задайте:
    - шрифт: Times New Roman, 14; полужирный, курсив;
    - абзац: выравнивание по левому краю; отступы слева и справа – 0 см; первая строка – нет; интервал перед – 6 пт, после – 3 пт, межстрочный – одинарный.
  - **МойСтиль3**, основанный на стиле «Заголовок 3», в котором задайте:
    - шрифт: Times New Roman, 12; полужирный, курсив;
    - абзац: выравнивание по левому краю; отступ слева – 1 см, справа – 0 см; первая строка – нет; интервал перед и после – 3 пт, межстрочный – одинарный.
  - **МойСтиль4**, основанный на стиле «Обычный», в котором задайте:
    - шрифт: Times New Roman, 12; курсив;
    - абзац: выравнивание по левому краю; отступ слева – 2 см, справа – 0 см; первая строка – 0,5 см; интервал перед и после – 0 пт, межстрочный – одинарный.
3. Чтобы применить созданные стили к заголовкам и абзацам набранного текста, перейдите в режим просмотра «Структура документа» (во вкладке меню **Вид**). Выделите строку с заголовком документа, затем выберите **МойСтиль1** в списке стилей. Далее для всех заголовков *глав* (I, II и III) выберите **МойСтиль2** и для всех заголовков *статей* – **МойСтиль3**. Оставшиеся абзацы оформите стилем **МойСтиль4**. Строку с датой (*19 апреля...*) выровняйте по правому краю и уменьшите размер шрифта до 10 пт.
4. Вернитесь в режим просмотра «Разметка страницы» (вкладка **Вид**). Добавьте **Маркированный список** (кнопка  на вкладке меню **Главная**) в статьи 2 и 12. Для этого выделите нужные строки, раскройте кнопку  (стрелочка справа от нее) и выберите нужный маркер. После этого кнопками  и  выровняйте списки, как в тексте.
5. В списке стилей найдите **МойСтиль4**, нажмите правой кнопкой мыши и выберите **Изменить**. Установите выравнивание *по ширине* и нажмите **ОК**. Убедитесь, что все абзацы, имеющие стиль **МойСтиль4**, выровнялись по ширине.
6. Самостоятельно добавьте **колонтитулы** и **автонумерацию** страниц (в верхнем колонтитуле напишите «Практическая работа №3», в нижнем – вашу фамилию и группу).
7. Принудительно начните новую страницу (**Ctrl + Enter**) и вставьте **Оглавление** (для этого на вкладке меню **Ссылки** раскройте стрелкой кнопку **Оглавление** и выберите одно из автоматических оглавлений).
8. Самостоятельно установите **защиту** документа:

- в режим Записи Исправлений. Попробуйте удалить или исправить некоторые участки текста, убедитесь, что все исправления фиксируются. Снимите защиту и правой кнопкой мыши отмените все исправления.
  - в режим Вставки Примечаний. Добавьте два любых примечания в документ.
  - в режим Только Чтение. Убедитесь, что редактирование документа запрещено.
9. Отключите защиту документа.

**Образец текста**

## **ИЗ ЗАКОНА «О ЗАНЯТОСТИ НАСЕЛЕНИЯ В РОССИЙСКОЙ ФЕДЕРАЦИИ»**

*19 апреля 1991 года      № 1032-1*

### **Глава I. ОБЩИЕ ПОЛОЖЕНИЯ**

#### **Статья 1. Занятость граждан**

*Занятость — это деятельность граждан, связанная с удовлетворением личных и общественных потребностей, не противоречащая законодательству Российской Федерации и приносящая, как правило, им заработок, трудовой доход (далее — заработок).*

#### **Статья 2. Занятые граждане**

*Занятыми считаются граждане:*

- *работающие по трудовому договору (контракту);*
- *занимающиеся предпринимательской деятельностью;*
- *проходящие военную службу;*
- *проходящие очный курс обучения в образовательных учреждениях.*

#### **Статья 3. Порядок и условия признания граждан безработными**

*Безработными признаются трудоспособные граждане, которые не имеют работы и заработка, зарегистрированы в органах службы занятости в целях поиска подходящей работы, ищут работу и готовы приступить к ней.*

### **Глава II. ПРАВА ГРАЖДАН В ОБЛАСТИ ЗАНЯТОСТИ**

#### **Статья 8. Право граждан на выбор места работы**

*Граждане имеют право на выбор места работы путем прямого обращения к работодателю, или путем бесплатного посредничества органов службы занятости, или с помощью других организаций по содействию в трудоустройстве населения.*

### **Глава III. ГАРАНТИИ ГОСУДАРСТВА В ОБЛАСТИ ЗАНЯТОСТИ**

#### **Статья 12. Гарантии государства в реализации права граждан на труд**

*Государство гарантирует гражданам Российской Федерации:*

- *свободу выбора рода деятельности, в том числе работы с разными режимами труда;*
- *охрану труда, правовую защиту от необоснованного увольнения;*
- *бесплатное содействие в подборе подходящей работы и трудоустройстве при посредничестве органов службы занятости.*

10. Создайте таблицу 2 x 9, используя меню «Вставить» команда «Таблица» или кнопку *Вставить таблицу* из панели инструментов нажатием и продвижением по таблице с помощью левой кнопки мыши

11. Измените ширину колонок по образцу табл. 2. Наведите стрелку мыши на вертикальный разделитель таблицы, при этом стрелка мыши примет вид разделителя; нажатием и продвижением разделителя левой кнопкой мыши задайте нужную ширину столбцов таблицы.
12. Изменение размеров ячеек можно производить, используя свойства таблиц. Для этого достаточно установить курсор внутри таблицы, в меню «Макет» или «Конструктор» выбрать команду Свойства таблицы. В появившемся диалоговом окне на вкладке Таблица можно изменить размер, выравнивание, обтекание. На вкладке Строка изменяется высота строк, а на вкладке Столбец - ширина столбцов, на вкладке Ячейка - размер ячейки. Автоподбор параметров таблицы можно произвести командой меню «Макет» команда «Автоподбор». Microsoft Word произведет автоматический подбор ширины столбцов или строк в зависимости от ширины листа и объема текста в каждой ячейке.

Таблица 2

Денежные параметры	Сумма, млрд долл.
Наличные деньги	<b>232</b>
Трансакционные депозиты	<b>563</b>
В том числе:	
вклады до востребования	<b>277</b>
прочие чековые депозиты	<b>286</b>
<i>Итого:</i>	<b>795</b>
Взаимные фонды денежного рынка	<b>318</b>
Депозитные счета денежного рынка	<b>485</b>
Сберегательные вклады	<b>410</b>
Срочные вклады	<b>1143</b>
Однодневные соглашения об обратном выкупе	<b>64</b>
Однодневные займы и прочее	<b>17</b>
<i>Итого:</i>	<b>3232</b>




6. Выделите первую строку таблицы (шапку) и задайте тип выравнивания абзаца - по центру.
7. Выделите второй столбец таблицы и задайте тип выравнивания абзаца - по центру.
8. Заполните таблицу, перемещаясь по ней с помощью клавиш [Tab], [Shift]-[Tab].
9. Добавьте в таблицу новую строку, для чего поместите курсор в правую ячейку нижней строки таблицы и нажмите клавишу [Tab] или воспользуйтесь в меню «Макет», команда «Добавить строки выше-ниже» предварительно установив курсор в любую ячейку нижней строки таблицы.
10. Произведите обрамление таблицы по образцу, используя команду «Границы страниц» в меню «Разметка страницы»
12. Проведите сортировку (по возрастанию) данных второй колонки таблицы, выделенных жирной линией.

## Практическая работа №12 Создание и редактирование графических и мультимедийных объектов средствами компьютерных презентаций


**Цель:** научиться создавать презентации в MS PowerPoint

### ХОД РАБОТЫ

1. Создание стандартной презентации.

- Запустите приложение MS PowerPoint. Щелкните  – “Создать”, в окне диалога Создание презентации выберите в “Образцах шаблонов” один из готовых шаблонов Презентации и нажмите кнопку “Создать”. Просмотрите слайды этой презентации, щёлкнув мышью на соответствующие слайды, в оглавлении слайдов (панель слева).
- Перейдите на вкладку меню “Вид” и выберите режим “Сортировщик слайдов”. Просмотрите состав слайдов этой презентации с помощью линейки прокрутки. На вкладке меню “Дизайн” выберите “Тему” слайдов. В окне выбора темы просмотрите различные дизайны и выберите понравившийся дизайн.
- Дважды щёлкните мышью по нему. Нажмите клавишу “Home”. Последовательно щёлкните мышью вкладку меню “Показ слайдов” – “С начала”. Осуществите демонстрацию на экране компьютера, переходя к следующему слайду щелчком мыши. Перетащите слайд № 2 в промежуток между слайдами 4, 5. После появления в этом промежутке тонкой вертикальной черты отпустите кнопку мыши. Нажмите клавишу “END”. Последовательно щёлкните мышью вкладку меню “Главная” – “Вырезать” (  ) для удаления слайда. Закройте презентацию (  – “Закреть”), на вопрос о сохранении ответьте Нет.

## 2. Создание титульного листа.

- Создайте новую презентацию (  – “Создать”). Если титульный слайд не появился, нажмите “Создать слайд” – “Титульный слайд”. В поле заголовка введите «Институт гуманитарных наук».
- В поле подзаголовка введите «Создан в 1992 году» щёлкните пункт меню “Вставка” – “Картинка”, в открывшемся справа окне нажмите “Начать”. Выберите рисунок и вставьте на слайд.
- Расположите рисунок в верхнем левом углу слайда, перемещая его мышью.

## 3. Создание файла со списком.

- Нажмите пункт “Главная” – “Создать слайд” (макет слайда – “Заголовок и объект”). Введите заголовок Специальности.
- В поле текста введите в столбик (после каждого слова нажимайте клавишу Enter): Юриспруденция, Экономика, Менеджмент, Лингвистика. Щёлкните пункт меню “Вставка” – “ Картинка ”. В открывшемся справа окне нажмите “Начать”. Выберите рисунок и вставьте его на слайд.
- Расположите рисунок в нижнем правом углу слайда, правее текста.

## 4. Создание слайда с таблицей.

- Нажмите пункт “Главная” – “Создать слайд” (макет слайда – “Заголовок и объект”). Введите заголовок Численность студентов. Щёлкните мышью по кнопке Таблица в центре слайда, в окне диалога “Вставка таблицы” установите столбцов 4, строк 5 и нажмите ОК. Заполните таблицу по образцу справа.

	1995	1996	1997
<b>Юриспруденция</b>	300	600	900
<b>Экономика</b>	200	400	600
<b>Менеджмент</b>	100	200	300
<b>Лингвистика</b>	50	100	150

## 5. Создание слайда с диаграммой.

- Нажмите пункт “Главная” – “Создать слайд” (макет слайда – “Заголовок и объект”). Введите заголовок Динамика роста численности. Щёлкните мышью по кнопке Диаграмма, в открывшемся окне выберите тип “График с маркерами” и нажмите ОК. В открывшейся таблице MS Excel заполните ячейки в соответствии с таблицей в предыдущем слайде и закройте окно Excel.



6. Шаблон оформления, анимация и показ слайдов.
  - Перейдите на вкладку “Дизайн” и выберите “Тему” слайдов. На вкладке меню “Анимация” для каждого объекта добавьте анимацию “Вход”. На вкладке меню “Переходы” в поле “Смена слайда” установите время для каждого слайда (4-6 секунд).
  - На вкладке меню “Показ слайдов” нажмите “С начала” для демонстрации презентации (или просто F5). Переход к следующему слайду в режиме просмотра осуществляется левой кнопкой мыши или пробелом, выход – Esc.
7. Создание собственной презентации.
  - Самостоятельно создайте свою презентацию, используя список, таблицу, диаграмму, анимацию.
  - Сохраните презентацию, закройте программу.

### Практическая работа №13 Обработка числовых данных. Применение итоговых функций

**Цель:** Изучить технологию обработки числовых данных и применения итоговых функций

#### ХОД РАБОТЫ

1. Создайте новую книгу MS Excel.
2. На «Листе 1» напишите тему и цель урока и переименуйте его в «Обработка числовых данных».
3. На следующем листе составьте таблицу из 10–15 записей по образцу:
  - **Должности:** Бухгалтер, Менеджер, Директор, Зам. директора, Кассир (текстовый формат);
  - **Зарплата** – от 12000 до 40000 р.;
  - **Премия** – от 20 до 60 процентов;
  - **Налог** – 13 процентов;
  - **Аванс** – 40 процентов;
  - Ячейки правее «Премия (%)» оставьте незаполненными, но установите денежный формат.

№	ФИО	Должность	Оклад (руб.)	Премия (%)	Премия (руб.)	Оклад + Премия	Налог (руб.)	Зарплата (руб.)	Аванс (руб.)
1	Иванов А.М.	Бухгалтер	16 750,00	22%					

4. В конце таблицы добавьте строку вида:

<b>Минимум:</b>			
<b>Максимум:</b>			
<b>В среднем:</b>			
<b>Итого:</b>			

5. Перед таблицей вставьте строки (формат ячеек с 13% и 40% – процентный):

<b>Налог:</b>	13%
<b>Аванс:</b>	40%

6. Назовите лист – «Задание 1».
7. Выполните Сортировку сотрудников по столбцу ФИО в алфавитном порядке. Восстановите нумерацию в столбце №.
8. Установите Формат ячеек: «Премия (%)» – Процентный, число десятичных знаков – 0;
9. «Оклад», «Премия (руб.)», «Оклад + Премия», «Налог (руб.)», «Зарплата», «Аванс» – Денежный, число десятичных знаков - 2, обозначение - нет.

### Формулы

10. Для выполнения простых математических действий (например, чтобы посчитать размер премии (ячейка F5), нужно умножить Оклад (E5) на Процент премии (D5)):

- выделите ячейку F5 и начните ввод формулы с символа “=”;
- нажмите на ячейку D5 с первым множителем (16 750,00 руб.);
- наберите символ “\*”;
- нажмите на ячейку E5 со вторым множителем (22%);
- нажмите **Enter** для завершения.

В ячейке F5 появится формула “=D5\*E5”, которая будет сразу вычислена.


11. Для быстрого заполнения ячеек таблицы однотипными формулами, скопируйте ячейку D7 и вставьте в остальные ячейки этого столбца. При копировании адреса ячеек автоматически меняются (в ячейку F6 формула копируется в виде “=D6\*E6”).

12. Чтобы запретить изменение адреса ячейки при копировании используйте **абсолютную адресацию**. Для изменения вида адресации установите курсор на адрес ячейки в формуле и нажмите клавишу F4. Адрес ячейки будет выглядеть так: \$D\$5.

13. Рассчитайте последовательно столбцы:

- «Премия» = «Оклад» \* «Процент Премии»;
- «Оклад + Премия»;
- «Налог» = «Оклад + Премия» \* «Процент Налога»;
- «Зарплата» = «Оклад + Премия» – «Налог»;
- «Аванс» = «Зарплата» \* «Процент Аванса».

При расчете **Аванса** и **Налога** используйте абсолютную адресацию для ячеек с 13% и 40%.

14. Для вставки формулы нажмите кнопку , выберите нужную **категорию** (напр. математические, статистические...), а затем **функцию**. В открывшемся окне редактирования аргументов функции впишите нужные **данные** (напр. диапазоны ячеек, отдельные ячейки, числа или текст) и нажмите **Ок**.

15. Используя формулы:

- **МАКС**(диапазон) – максимальное значение в диапазоне ячеек,
- **МИН**(диапазон) – минимальное значение в диапазоне ячеек,
- **СРЗНАЧ**(диапазон) – среднее значение в диапазоне ячеек,
- **СУММ**(диапазон) – сумма значений диапазона ячеек

заполните ячейки в строках под таблицей.

### Практическая работа №14 Оформление таблиц. Построение графиков функций и диаграмм различных видов

**Цель:** Получить практические навыки оформления таблиц. Построение графиков функций и диаграмм различных видов.

#### ХОД РАБОТЫ

1. Составьте прайс-лист по образцу:

Прайс-лист магазина «Канцтовары»		
		<b>28.09.16</b>
	Курс доллара	63.5.
Наименование товара	Цена в у.е.	Цена в руб.
Тетрадь в клеточку	\$0,20	

Тетрадь в линейчку	\$0,20	
Пенал	\$2,00	
Ручка	\$0,50	
Карандаш	\$0,20	
Линейка	\$0,30	
Итого		

- Выделите ячейку В1 и введите в нее заголовок таблицы «**Прайс-лист магазина Канцтовары**»
  - В ячейку С2 введите функцию «**СЕГОДНЯ**» (Поставьте знак «=», Нажмите кнопку «fx» на панели инструментов. В поле «**КАТЕГОРИЯ**» выберите «**Дата и Время**». В нижнем поле выберите функцию «**Сегодня**»).
  - В ячейку В3 введите слова «Курс доллара», в С3 – курс доллара на сегодняшний день.
  - К ячейке С3 примените денежный формат (меню «**Главная**» блок «**Выравнивание**» кнопка «**Число**»).
  - В ячейки А5:В5 введите заголовки столбцов таблицы. Выделите их и примените полужирный стиль начертания и более крупный шрифт (14).
  - В ячейки А6:А12 и В6:В12 введите данные.
  - В ячейку С6 введите **формулу: = В6\*\$С\$3**. (\$ означает, что используется абсолютная ссылка, которая позволяет привязать ячейку к формуле).
  - Выделите ячейку С6 и протяните за маркер заполнения вниз до ячейки С13.
  - Выделите диапазон ячеек С6:С13 и примените к ним денежный формат.
  - Выделите заголовок – ячейки В1:С1 и выполните команду «**Выравнивание**» и установите переключатель «**Центрировать по выделению**» (Горизонтальное выравнивание), «**Переносить по словам**». Увеличьте шрифт заголовка (16).
  - Измените название ЛИСТ1 на Прайс-лист.
2. Создайте новую книгу MS Excel.
  3. Переименуйте «Лист 1» в «**Диаграммы**».
  4. Откройте вкладку меню Вставка и запишите в столбик типы диаграмм, доступные в группе Диаграммы.
  5. На следующем листе (название – «**Задание 1**») постройте таблицу (см. рисунок).
  6. Для построения диаграммы (тип диаграммы на рисунке – Гистограмма) выделите область таблицы данных (область В4:С7), нажмите кнопку Гистограмма, выберите подтип, схожий с рисунком.
  7. Щелкните область диаграммы. Перейдите на вкладку меню Работа с диаграммами – Макет. Используя группу Подписи, добавьте подписи рядов (справа от диаграммы, 2015 и 2016), подписи данных (оси X и Y), название диаграммы (сверху от диаграммы).
  8. Для изменения формата отдельных элементов диаграммы (цвет, заливка, границы...) перейдите на вкладку меню Работа с диаграммами – Формат. В группе Текущий фрагмент щелкните стрелку рядом с полем Элементы диаграммы и выберите нужный. Измените цвета столбцов диаграммы, заливку, цвета и толщину границ.
  9. Самостоятельно создайте еще 6 листов (названия: «**Задание 2**», «**Задание 3**»...«**Задание 7**»).
  10. На каждом листе по образцу «**Задания 1**» составьте таблицы с данными в соответствии с рисунком.
  11. Постройте по этим данным соответствующие диаграммы.
  12. Измените формат диаграмм в соответствии с рисунками (3-6).

	A	B	C	D
1	<b>Продажа лёгких закусок</b>			
2				
3		<b>Закуска</b>	<b>Продажи (%)</b>	
4		Сэндвичи	40%	
5		Салаты	21%	
6		Супы	15%	
7		Напитки	9%	
8		Десерты	15%	
9				



Рис. 3

	A	B	C	D	E
1					
2		<b>Сорта</b>			
3		<b>Сорт А</b>	<b>Сорт В</b>	<b>Сорт С</b>	
4		Витамин А	45%	100%	45%
5		Витамин В1	25%	25%	65%
6		Витамин В2	65%	35%	85%
7		Витамин С	12%	84%	74%
8		Витамин D	35%	95%	96%
9		Витамин E	48%	22%	65%

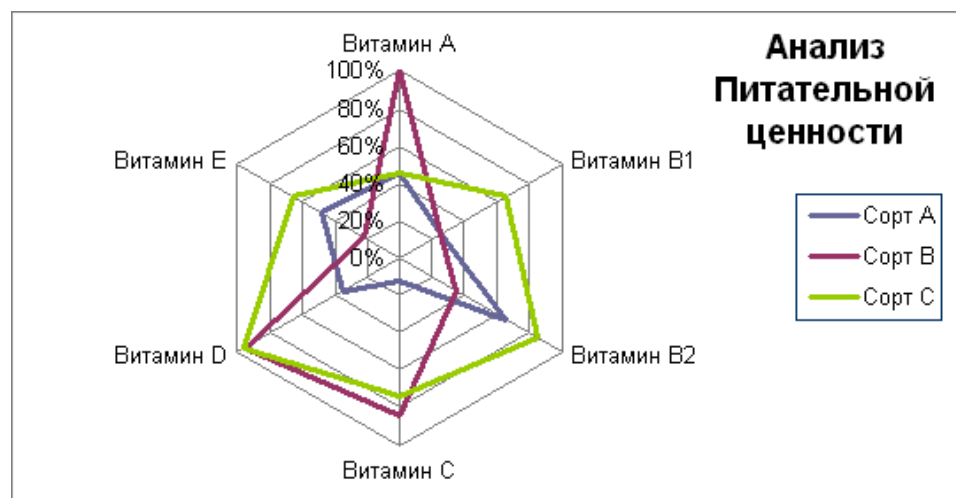


Рис. 4

	А	В	С
1			
2		<b>Страна</b>	<b>сумма</b>
3		Япония	122132
4		Европа	115456
5		США	45215



Рис. 5

	А	В	С	Д	Е	Ф
1						
2		<b>Страна</b>	<b>Кв.1</b>	<b>Кв.2</b>	<b>Кв.3</b>	<b>Кв.4</b>
3		Европа	13164	7897	454461	454
4		США	11561	4564	45456	12123
5		Япония	456465	12132	1131	45612

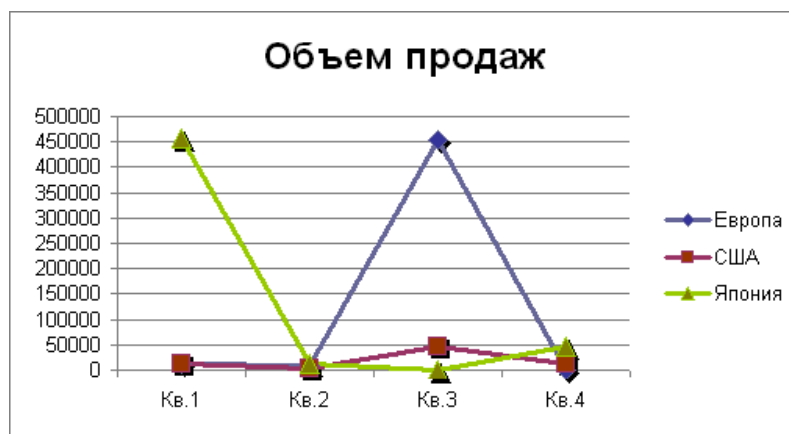


Рис. 6

### Практическая работа №15 Решение расчетных задач. Решение уравнений

**Цель:** Получить практические навыки решения расчетных задач и решения уравнений.

#### ХОД РАБОТЫ

1. Рассчитайте ведомость выполнения плана товарооборота по форме:

№	Месяц	Отчетный год			Отклонение от плана
		план	фактически	выполнение, %	
	М	Р	Ф	У	О
1	Январь	7 800,00 р.	8 500,00 р.		
2	Февраль	3 560,00 р.	2 700,00 р.		
3	Март	8 900,00 р.	7 800,00 р.		

4	Апрель	5 460,00 р.	4 590,00 р.		
5	Май	6 570,00 р.	7 650,00 р.		
6	Июнь	6 540,00 р.	5 670,00 р.		
7	Июль	4 900,00 р.	5 430,00 р.		
8	Август	7 890,00 р.	8 700,00 р.		
9	Сентябрь	6 540,00 р.	6 500,00 р.		
10	Октябрь	6 540,00 р.	6 570,00 р.		
11	Ноябрь	6 540,00 р.	6 520,00 р.		
12	Декабрь	8 900,00 р.	10 000,00 р.		

–Заполнение столбца №, можно осуществить через автозаполнение (в ячейку А3 ввести цифру 1, в ячейку А4 ввести цифру 2, затем выделить эти 2 ячейки и потянуть за маркер автозаполнения до цифры 12».

–Заполнение столбца «Месяц» также осуществляется с помощью маркера автозаполнения (напечатать 1 месяц – январь, а затем потянуть за левый нижний угол ячейки вниз до «Декабря»).

–Значения столбцов V и O вычисляются по формулам:  $V=F / P$ ;  $O_i=F - P$

–Переименуйте ЛИСТ2 в Ведомость (щелкнув правой кнопкой мыши по листу, выберите команду «Переименовать»).

–Сохраните таблицу в своей папке под именем Практическая работа 1

2. Решите систему  $\begin{cases} y - x^2 = 0, \\ y + 2x = 1 \end{cases}$  графическим способом.

–Для решения воспользуйтесь диаграммой, на которой отобразите графики обеих функций.

–Заполните ячейки А2:А22 числами от -5 до 5 с шагом 0,5. (в ячейку А2 заносим число -5, в ячейку А3 – число -4,5, выделяем ячейки А2 и А3, установим курсор мыши на правый нижний угол рамки (указатель примет форму черного крестика) и тяните курсор вниз, пока последнее значение не станет равным 5).

–При заполнении столбца В в ячейку В2 введите формулу =А2\*А2, которую затем копируйте до ячейки В22. (протяните формулу за правый нижний угол).

–При заполнении столбца С в ячейку С2 введите формулу =1-2\*А2, скопируйте ее до ячейки С22.

–Выделите блок с данными, с помощью Мастера диаграмм выберем тип диаграммы «Точечная» и постройте графики функций. Координаты точек пересечения графиков – решения системы.  $\{(-2,5; 6); (0,5; 0)\}$

–Получены приближенные значения решений. Чем меньше шаг, тем точнее значение координат точек пересечения.

## Практическая работа №16 Создание базовых таблиц и межтабличных связей

**Цель:** Изучить технологию создания таблиц и межтабличных связей.

### ХОД РАБОТЫ

1. Откройте базу данных «Список сотрудников филиала».
2. Создайте таблицу «Подразделения» в Режиме таблицы. Для этого в меню «Создание» выберите кнопку «Таблица». В появившейся таблице сделайте следующее:

- Добавьте два поля – Поле 1 и Поле 2, выполнив команду через контекстное меню.
  - Переименуйте «Поле 1» на «Подразделение». Для этого поставьте курсор в любую ячейку столбца «Поля 1» и выполните команду «Переименовать» столбец из контекстного меню. Или щелкните два раза по имени поля, удалите старое название и впечатайте новое.
  - Переименуйте аналогично «Поле 2» на «Отделение».
3. Сохраните таблицу с именем «Подразделение», щелкнув по кнопке «Сохранить».
  4. Перейдите в режим «**Конструктор**» и удалите строку с ключевым словом Счетчик. Посмотрите как заданы поля. Сделайте поле «Отделение» ключевым, поместив курсор на имя этого поля и щелкнув по кнопке – «**Ключевое поле**». Тип данных поля задайте *текстовым*.
  5. Перейдите в «Режим таблицы» и заполните таблицу «Подразделение» произвольными данными.
  6. Закройте таблицу «Подразделение», сохранив все изменения.
  7. Создайте схему данных, т.е. установите связи между таблицами.

Щелкните по кнопке - Схема данных в меню «Работа с базами данных». В окне «Отобразить таблицу» выделите таблицу «Список сотрудников филиала» и щелкните по кнопке «Добавить». Также добавьте таблицы «Подразделение». В окне «Схема данных» появиться условный вид этих таблиц. Закройте окно «Добавление таблицы».

–Поставьте мышку на имя поля «Подразделение» в таблице «Список сотрудников филиала», и не отпуская кнопку мыши перетащите его на поле «Отдел» таблицы «Подразделения». Отпустите мышку. Появиться диалоговое окно «Связи», в котором включите значки «Обеспечение целостности данных», «Каскадное обновление связанных полей» и «Каскадное удаление связанных полей». Щелкните по кнопке «Создать». Появиться связь «один-ко-многим».

–Закройте схему данных, сохранив ее.

–Закройте таблицу и базу данных.

### **Практическая работа №17 Создание запросов (на выборку, с параметром, итогового запроса)**

**Цель:** Изучить технологию создания запросов: на выборку, с параметром, итогового запроса

#### **Теоретический материал**

Запросы состоят из ряда условий, каждое условие состоит из трех элементов:

1. поле, которое используется для сравнения;
2. оператор, описывающий тип сравнения;
3. величина, с которой должно сравниваться значение поля.

#### Выражения и операторы, применяемые в условиях отбора.

Выражения и операторы	Описание выражений и операторов
Числа	Вводятся без ограничений
Текст	Должен быть заключен в кавычки
Даты	Ограничиваются с двух сторон символами # (например, #01.02.02#)

*; +; -; /; ^	Арифметические операторы, связывающие выражения
<; <=; >; >=; =; <>	Операторы сравнения
And (И); Not (Нет); Or (Или)	Логические операторы
Like	Используется для логики замены в выражениях
In	Для определения, содержится ли элемент данных в списке значений
Between... And...	Для выбора значений из определенного интервала
?	Заменяет один символ (букву или цифру)
*	Заменяет несколько символов

### ХОД РАБОТЫ

- Откройте базу данных «Список сотрудников филиала».
- Создайте запрос на выборку сотрудников, которые работают в основном отделе с помощью «Мастера запросов».
  - На панели инструментов выберите команду «Мастер запросов».
  - В появившемся диалоговом окне выберите «Простой запрос» и щелкните по кнопке «ОК».
  - В следующем окне выберите таблицу, по которой строится запрос «Список сотрудников», и те поля, которые участвуют в запросе. Перенесите их в правую часть окна, нажмите «Далее». В следующем окне тоже нажмите «Далее».
  - В другом окне дайте название запроса «Отдел» и нажмите «Готово».
  - Появится таблица «Отдел», в которой отражены фамилии всех сотрудников определенного отдела.
- С помощью «Конструктора запросов» создайте запрос на выборку по таблице «Список сотрудников».
  - Щелкните по таблице «Список сотрудников, зайдите в меню «Создание», выберите команду «Конструктор запросов».
  - Добавьте нужную таблицу в поле запроса. Выделите её в списке и щелкните по кнопке «Добавить». Закройте окно «Добавление таблицы».
  - Выберите сотрудников, чьи фамилии начинаются на букву «С» и которые проживают в Нефтеюганске. Для этого:
    - добавьте в строку «Поле» два поля «Фамилия» и «Город»;
    - в строке «Условия отбора» в первом столбце укажите значение **Like “С \* ”**, а во втором столбце с названием «Город» - **«Нефтеюганск»**;
    - закройте запрос, сохранив его под названием **“Фамилия-город”** (у вас должны остаться в списке сотрудники, проживающие в Нефтеюганске).
- Открыть базу данных «Сотрудники филиала». Составьте запрос на базе имеющихся таблиц. Выберите сотрудников, которые проживают в Нижневартовске и у которых стаж работы 5 лет. Озаглавьте «Запрос по стажу». Используйте «Конструктор запросов».
  - В меню «Создание» выберите «Конструктор запросов».
  - Добавьте все таблицы в поле запроса. Закройте окно «Добавление таблицы».
  - В первый столбец в строку «Поле» перетащите из первой таблицы с помощью мышки «Фамилия», «Город» «Стаж».
  - В поле «Условия отбора» в столбце «Город» введите город «Нижневартовск», в столбец «Стаж» введите 5«».
  - Сохраните запрос под именем **«Запрос по стажу»**.
  - Отсортируйте по возрастанию



- Откройте запрос и проверьте результат проделанной работы.
5. Выполните запрос на создание новой таблицы, в которой должны быть поля «Фамилия», «Имя», «Город» и «Отдел».
    - В меню «Создание» выберите «Конструктор запросов».
    - Добавьте все таблицы из списка окна «Добавление таблицы». Закройте это окно.
    - В первую строчку «Поле» из первой таблицы перенесите в первый столбец поля «Фамилия», во второй «Имя» и в третий «Город», в четвертый столбец перетащите поле «Отдел» Закройте запрос, сохранив его с именем «Запрос по отделу».
  6. Самостоятельно создайте запрос на поиск информации в базе данных, используя таблицу «Подразделение»

### **Практическая работа №18 Создание форм различными способами**

**Цель:** научиться создавать формы различными способами

#### **ХОД РАБОТЫ**

1. Открыть базу данных «Список сотрудников филиала». Создайте форму с помощью **«Мастера форм»**.
  - Откройте вкладку «Создание», меню «Формы».
  - Выберите «Мастер форм» и таблицу «Список сотрудников филиала».
  - Выберите нужные поля, которые будут участвовать в форме, нажмите кнопку «Далее».
  - В новом окне выберите поля для группировки так, чтобы сначала было указано поле «Фамилия», нажмите кнопку «Далее».
  - На этом шаге отсортируйте данные по алфавиту, нажмите кнопку «Далее».
  - Выберите вид макета Ступенчатый и щелкните по кнопке «Далее».
  - Выберите стиль отчета: Открытая и щелкните по кнопке «Далее».
  - Задайте имя формы: «Сотрудники фирмы» и щелкните по кнопке «Готово». Вы попадете в режим ввода данных в форму.
  - Закройте форму согласившись с сохранением.
2. Создайте самостоятельно пользовательскую форму с помощью «Конструктора форм». Изучите работу с «Конструктором форм» используя справочную систему программы.

### **Практическая работа №19 Создание отчетов**

**Цель:** научиться создавать отчеты по таблицам базы данных

**Отчет** – это гибкое и эффективное средство для организации просмотра и распечатки итоговой информации. В отчете можно получить результаты сложных расчетов, статистических сравнений, а также поместить в него рисунки и диаграммы. Пользователь имеет возможность разработать отчет самостоятельно (в режиме *Конструктора*) или создать отчет с помощью *Мастера*, т.е. полуавтоматически.

#### **ХОД РАБОТЫ**

1. Открыть базу данных «Список сотрудников филиала». Создайте отчет с помощью **«Мастера отчетов»**.
  - Откройте вкладку «Создание», меню «Отчеты».
  - Выберите «Мастер отчетов» и таблицу «список сотрудников филиала».

- Выберите нужные поля, которые будут участвовать в отчете, нажмите кнопку «Далее».
  - В новом окне выберите поля для группировки так, чтобы сначала было указано поле «Фамилия», нажмите кнопку «Далее».
  - На этом шаге отсортируйте данные по алфавиту, нажмите кнопку «Далее».
  - Выберите вид макета Ступенчатый и щелкните по кнопке «Далее».
  - Выберите стиль отчета: Открытая и щелкните по кнопке «Далее».
  - Задайте имя отчета: «Отчет1» и щелкните по кнопке «Готово». Вы попадете в режим просмотра отчета.
  - Закройте отчет согласившись с сохранением.
2. Составьте еще два отчета по запросам – «Запрос по стажу» и «Подразделение», выбирая из разных макетов: **блок**; **структура**, выбирая из разных стилей. Сохраните отчеты под именами «Отчет 2» и «Отчет 3».
  3. Создайте **Пустой отчет** в столбец на базе таблицы «Список сотрудников филиала» и сохраните его с именем «Сотрудники».
    - С помощью Конструктора измените цвет букв заголовка, их размер и шрифт.
  4. Создайте почтовые наклейки.
    - Откройте вкладку «Создание», меню «Отчеты».
    - Выберите таблицу «Список сотрудник филиала», команда «Наклейки».
    - В следующем окне щелкните по кнопке «Далее».
    - В следующем окне выберите шрифт, размер шрифта, насыщенность и цвет, вновь щелкните по кнопке «Далее».
    - В следующем окне создайте прототип наклейки, напечатав слово СОТРУДНИК и выбрав соответствующие поля, щелкните по кнопке «Далее».
    - В следующем окне укажите поля для сортировки (Фамилия, Имя), щелкните по кнопке «Далее».
    - Введите имя отчета «Наклейки» и щелкните по кнопке «Готово».
    - Просмотрите Наклейки, сохраните выйдите из программы.

## **Практическая работа №20 Основные элементы языка Паскаль. Разделы описаний**

**Цель:** Изучить основные элементы языка Паскаль. Разделы описаний

### **ХОД РАБОТЫ**

1. Изучить теоретический материал
2. Заполнить таблицу 3.

### **Теоретический материал**

#### **Основные элементы языка Паскаль**

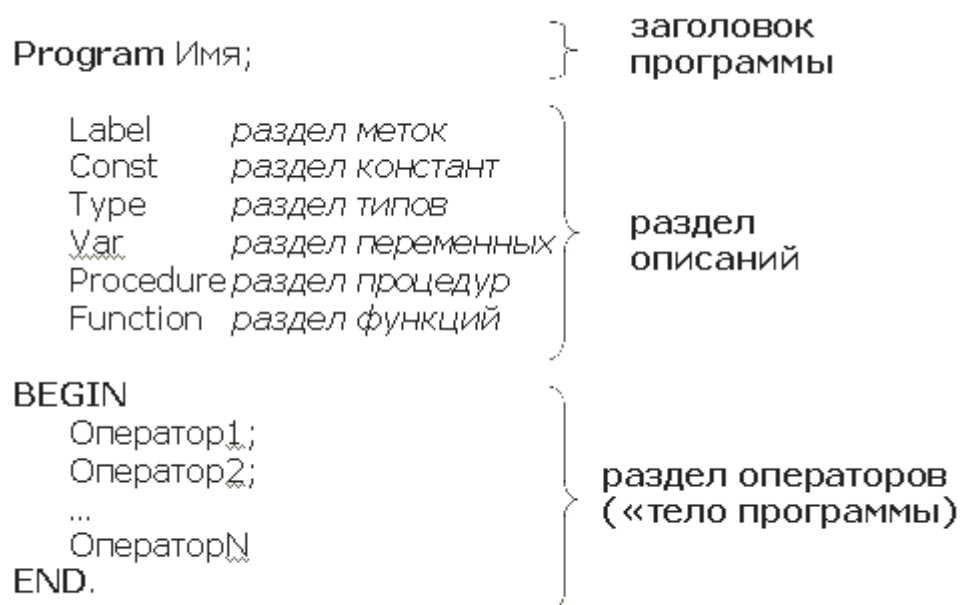
Программа реализует алгоритм решения задачи. В ней программист записывает последовательность действий, выполняемых над определенными данными с помощью определенных операций для реализации заданной цели. Основные характеристики программы: точность полученного результата, время выполнения и объем требуемой памяти.

Программа на языке Паскаль состоит из строк. Набор текста программы осуществляется с помощью встроенного редактора текстов системы программирования Турбо Паскаль или любого другого редактора формата DOS.

Максимальный размер программы ограничен. Компилятор позволяет обрабатывать программы и библиотечные модули, в которых объем данных и генерируемый машинный

код не превышают 64 Кбайт каждый. Если программа требует большего количества памяти, следует использовать библиотечные модули (.TPU-файлы) или оверлейные структуры.

### Синтаксическая диаграмма блока:



Заголовок программы несет чисто смысловую нагрузку и может отсутствовать, однако рекомендуется всегда его записывать (на латинском регистре) для быстрого распознавания нужной программы среди листингов других программ. После заголовка следует программный блок, состоящий в общем случае из семи разделов:

- списка имен подключаемых библиотечных *модулей* (он определяется зарезервированным словом **uses**);
- описания *меток*;
- описания *констант*;
- определения *типов данных*;
- описания *переменных*;
- описания *процедур и функций*;
- операторов (**begin...end**).

Любой раздел, кроме раздела операторов, может отсутствовать. Разделы описаний (кроме **uses**, который всегда расположен после заголовка программы) могут встречаться в программе любое количество раз и следовать в произвольном порядке. Главное, чтобы все описания объектов программы были сделаны до того, как они будут использованы.

### РАЗДЕЛ USES

Этот раздел состоит из зарезервированного слова **uses** и списка имен подключаемых стандартных и пользовательских библиотечных модулей.

Формат:

**uses** <имя1>, <имя2>, ... ;

Пример.

**uses** Crt;

## РАЗДЕЛ ОПИСАНИЯ МЕТОК

Перед любым оператором языка Паскаль можно поставить метку, что позволяет выполнить прямой переход на этот оператор с помощью оператора перехода `go to` из любого места программы.

Примечание. Нельзя выполнять переход на оператор в теле цикла, внутри составного оператора.

Метка состоит из имени и следующего за ним двоеточия. Именем может служить идентификатор или цифра. Максимальная длина имени метки ограничена 127 символами. Перед употреблением метка должна быть описана. Раздел описания меток начинается зарезервированным словом `label` (метка), за которым следуют имена меток, разделенные запятыми. За последним именем ставится точка с запятой.

Формат:

```
label <имя, ...>;
```

Пример.

```
label
```

```
Metka1, Metka2;
```

После записи метки в разделе операторов следует двоеточие, показывающее компилятору, что идентификатор используется как метка:

```
label
```

```
M1, M2;    {Описание меток}
```

```
begin
```

```
...
```

```
M1: <оператор> {Использование M1 в разделе операторов}
```

```
...
```

```
M2: <оператор> {Использование M2 в разделе операторов}
```

```
end.
```

Если метка описана, но в разделе операторов не используется, то ошибки при этом не возникает, т.е. метки можно описывать и применять по мере расширения программы.

## РАЗДЕЛ ОПИСАНИЯ КОНСТАНТ

В разделе описания констант производится присваивание идентификаторам констант постоянных значений. Раздел начинается зарезервированным словом `const`, за которым следует ряд выражений, присваивающих идентификаторам постоянные числовые или строковые значения. Выражения присваивания отделяются друг от друга

точкой с запятой. Формат:

```
const <идентификатор> = <значение>;
```

Пример.

```
const
```

```
Maxind: word = 100; {Типизированная константа}
```

```
Name = 'Петя'; {Строковая константа}
```

```
Code = $124;    {Константа — шестнадцатеричное значение}
```

## РАЗДЕЛ ОПИСАНИЯ ТИПОВ ДАННЫХ

Тип данных может быть либо описан непосредственно в разделе описания переменных, либо определяться идентификатором типа. Стандартные типы не требуют описания в отличие от типов, образованных пользователем. Строго говоря, синтаксис языка Паскаль не требует обязательного определения идентификатора типа и в последнем случае, так как тип можно задать перечислением в разделе описания переменных. Выбор описания типа зависит, таким образом, только от программиста и специфики программы.

Раздел описания типов данных начинается зарезервированным словом `type`, за которым следуют одно или несколько определений типов, разделенных точкой с запятой.

Формат записи:

```
type <имя типа> = <значения типа>;
```

Пример.

```
type
```

```
LatLetter = ('A'..'z'); Days = 1. . 31;
```

```
Matr = array[1..10] of integer;
```

Каждое описание задает множество значений и связывает с этим множеством некоторое имя типа. Например, в данном описании тип `LatLetter` определяет множество букв латинского алфавита. `Days` — множество целых чисел от 1 до 31, `Matr` — массив из 10 целых чисел.

## РАЗДЕЛ ОПИСАНИЯ ПЕРЕМЕННЫХ

Каждая встречающаяся в программе переменная должна быть описана. Описание обязательно предшествует использованию переменной. Раздел описания переменных

начинается зарезервированным словом `var` (`variable` — переменная), затем через запятую перечисляются имена переменных и через двоеточие следуют их тип и точка с запятой. Формат:

```
var
```

```
<идентификатор, . . . > : <тип>;
```

В рассматриваемом примере программы три переменных `A`, `B` и `Proizved` могут принимать целочисленные значения, описаны следующим образом:

```
var
```

```
A, B, Proizved : integer;
```

## РАЗДЕЛ ОПИСАНИЯ ПРОЦЕДУР И ФУНКЦИЙ

В общем случае подпрограмма имеет ту же структуру, что и программа. Для описания подпрограмм используются зарезервированные слова `procedure` и `function`,

которые записываются в начале подпрограммы. Формат процедуры:

```
procedure <имя процедуры> {<параметры>} ;
```

```
<разделы описаний>
```

```
<раздел операторов> end;
```

Формат функции:

```
function <имя функции> {<параметры>} : <тип результата>;
```

```
<разделы описаний>
```

```
<раздел операторов> end;
```

## РАЗДЕЛ ОПЕРАТОРОВ

В программе на языке Паскаль раздел операторов является основным, так как именно в нем с предварительно описанными переменными, константами, значениями

функций выполняются действия, позволяющие получить результат, ради которого создавалась программа.

Раздел операторов начинается зарезервированным словом `begin` (начало), далее следуют операторы языка, отделенные друг от друга точкой с запятой. Завершает раздел зарезервированное слово `end` (конец) с точкой.

Например:

```
begin {Начало программы}
```

```
Write ('Введите значение целого числа A >'); {Вывод запроса на экран}
```

```

Readln (A); {Ввод значения A с клавиатуры} Write ('Введите значение целого
числа B >'); Readln (B);
Proizved := A * B; {Вычисление переменной Proizved}
Write ('Произведение чисел ',A,' и ',B,' = ',Proizved); {Вывод ответа}
end. {Конец программы}

```

Операторы выполняются строго последовательно в том порядке, в котором они записаны в тексте программы в соответствии с синтаксисом и правилами пунктуации.

Слова `begin` и `end` являются аналогом открывающей и закрывающей скобки в обычных арифметических выражениях.

Таблица 3

Наименование раздела	Формат процедуры

### Практическая работа №21 Первые программы. Ввод и вывод данных

**Цель:** Получить навыки написания простых программ, ввода – данных

#### ХОД РАБОТЫ:

1. Набрать программу ввода трех произвольных целых чисел. Записать программу в тетрадь.

```

Program Primer;
Var k, l, m: integer;
Begin
  writeln ('введите целые числа k, l, m:');
  Readln (k, l, m);
  Readln;
End.

```

2. Замените в тексте многоточия изображением целых чисел (Integer) в строках `A:= ... ; B:= ... ;` и изображениями вещественных чисел (Real) в строках `C := ... ; D := ... ;`. Записать программу в тетрадь.

При ошибках в записи чисел транслятор об этом сообщит. Замените имя A (во всем тексте) на другой идентификатор. Ошибка в записи также будет отмечена транслятором. Поупражняйтесь в выборе имен. Два последних оператора позволяют определить диапазон представления целых и вещественных чисел. Определите этот диапазон, используя сообщение транслятора о “переполнении”.

```

Program XXX;
var
  A,B,S : Integer;
  C,D,R : Real;
begin

```

```

A := ... ;
B := ... ;
C := ... ;
D := ... ;
S := A+B;
R := C+D;
  writeln ('S=', S);
  writeln ('R=', R);
Readln;
end.

```

3. Набрать программу вычисления суммы двух целых чисел.

$$S = a + b$$

В программе использовать построчный ввод чисел.

```

Program Sum;
Var S, a, b: integer;
Begin
  writeln ('введите a:');
Readln (a);
  writeln ('введите b:');
Readln (b);
S:=a+b;
  writeln ('S=', S);
Readln;
End.

```

4. Составить одну программу вычисления следующих выражений (переменные a,b- целые и вещественные):

$$y = a + b$$

$$y = a - b$$

$$y = a^2$$

$$y = b^2$$

$$y = a^2 - b^2$$

Заполнить таблицу значениями.

a	b	$y = a + b$	$y = a - b$	$y = a^2$	$y = b^2$	$y = a^2 - b^2$
2	1					
4	3					
6	5					
5	3					
5	4					

5. Самостоятельно рассчитать площадь треугольника по формуле Герона (используя стандартные математические функции):

$$S = \sqrt{(p \cdot (p-a) \cdot (p-b) \cdot (p-c))}$$

, где  $p=(a+b+c)/2$  - полупериметр.

Исходные данные- a, b, c - стороны треугольника.

- 1) a, b, c – целые числа. S, p – вещественные числа. Не использовать форматированный вывод переменных.
- 2) a, b, c, S, p – вещественные числа. Использовать форматированный вывод переменных `Writeln ('s=', s:4:2)`

6. Ввести с клавиатуры два целых числа, найти отношение  $x = \frac{n1}{n2}$  первого числа ко второму и вывести полученный результат на экран. Записать программу в тетрадь.

```

Program Primer;
Var n1, n2: integer;      {вводимые числа}
      x: real;
Begin
  writeln ('n1='); {сообщить о вводе n1}
  Readln (n1); {ввести n1}
  writeln ('n2='); {сообщить о вводе n2}
  Readln (n2); {ввести n2}
  x:=n1/n2; {найти результат}
  writeln ('n1/n2=', x); {вывести результат}
Readln;
End.

```

7. Составить программу вычисления функции  $y = e^{2t}$ . Записать программу в тетрадь.

```

Program First;
Var t: integer;
      y: real;
begin
  t:=2;
  y:= exp(2*t);
  write ('y=', y);
  Readln;
end.

```

8. Составить программу вычисления площади прямоугольника произвольных размеров. Записать программу в тетрадь.

```

Program First;
Var a, b, S: real;
begin
  writeln ('введите длину прямоугольника');
  Readln (a);
  writeln ('введите ширину прямоугольника');
  Readln (b);
  S:=a*b;
  writeln ('площадь прямоугольника с шириной ', a:8:2, ' и длиной ', b:8:2, ' равна ', S:8:2);
  Readln;

```



**End.**

## Практическая работа №22 Оператор условия IF, оператор выбора Case

### Цель:

- 1) Освоить простейшую структуру программы с использованием оператора условия, оператора цикла
- 2) Получить практические навыки работы в интерактивном (диалоговом) режиме.
- 3) Получить практические навыки работы над ошибками.

### Теоретический материал

В программе разветвляющейся структуры имеется один или несколько условных операторов. Условный оператор в языке Паскаль имеет формат:

if <условие> then <оператор1> else <оператор2>; (полная форма) или  
if <условие> <оператор1>; (сокращенная форма).

ПРИМЕР: If a>=b then x:=a\*b else x:=a+b;

Точка с запятой перед зарезервированным словом else **не ставится**.

При выполнении условия выполняется оператор1, при невыполнении - оператор2 (при полной форме условного оператора). Для неполной формы условного оператора при выполнении условия выполняется оператор1, в противном случае оператор1 пропускается и выполняется оператор, следующий за условным оператором.

Оператор1 и оператор2 могут представлять простые операторы (один оператор), в этом случае они не заключаются в операторные скобки. Если же оператор1 и/или оператор2 представляют составной оператор (несколько операторов), то их нужно заключить в операторные скобки begin ... end.

Если количество разветвлений программы больше двух, то используется оператор вариантов Case, который является более общим случаем условного оператора. Оператор Case выполняет один из нескольких операторов в зависимости от значения селектора. Если селектор не совпадает ни с одним из значений, то выполняется оператор после else или следующий после case, если else отсутствует.

Селектор может быть целого, символьного, перечислимого, интервального типа или выражением.

### ХОД РАБОТЫ:

1. Составить программу: **если**  $x > 0$  **то**  $y := \sin(x)$  . Записать программу в тетрадь.

```
Program Wetw;  
Var x, y: real;  
Begin  
  writeln ('Введите x');  
  Readln (x);  
  If x>0  
  then y:=sin(x)  
  writeln (y);  
  Readln;  
End.
```

2. Ввести два числа с клавиатуры  $a$  и  $b$ .  
Проверить условие: **если**  $a \geq b$  **то**  $x := a * b$  **иначе**  $x := a + b$ .  
Записать программу в тетрадь.

```
Program Wetw;  
Var a, b, x: real;  
Begin  
  writeln ('Введите a');  
  Readln (a);  
  writeln ('Введите b');  
  Readln (b);  
  If a >= b then x := a * b  
  else x := a + b;  
  writeln ('x=', x:8:2);  
  Readln;  
End.
```

3. Ввести два числа с клавиатуры. Найти максимальное из двух вещественных чисел. Записать программу в тетрадь.

```
Program Wetw;  
Var x, y, max: real;  
Begin  
  writeln ('Введите x, y');  
  Readln (x, y);  
  If x >= y  
  then max := x  
  else  
  max := y;  
  writeln ('Введены два числа x и y');  
  writeln ('max=', max:6:2);  
  Readln;  
End.
```

4. Ввести два числа с клавиатуры. Найти максимальное из трех целых чисел. Записать программу в тетрадь.

```
Program Wetwlenimax;  
Var x, y, z, max: real;  
Begin  
  writeln ('Введите x, y, z');  
  Readln (x, y, z);  
  If x >= y  
  then max := x  
  else  
  max := y;  
  If x >= y;
```

```

z >= max then max := z;
  writeln ('Введены три числа x, y, z');
  writeln ('max=', max:6:2);
Readln;
End.

```

5. Набрать программу вычисления  $y = |x|$ . Записать программу в тетрадь.

$$|x| = \begin{cases} x, & \text{если } x \geq 0, \\ -x, & \text{если } x < 0. \end{cases}$$

```

Program Wetw;
Var x, y: real;
Begin
  writeln ('введите x');
  Readln (x);
  If x >= 0 then
    y := x
  else
    y := -x;
  writeln ('y=', y:8:2);
  Readln;
End.

```

6. Определить максимальное и минимальное значение из трех различных вещественных чисел. Записать программу в тетрадь.

```

Program Wetw;
Var A, B, C, max, min: real;
Begin
  writeln ('введите A, B, C');
  Readln (A, B, C);
  IF (A > B) AND (A > C) THEN MAX = A;
  IF (B > C) AND (B > A) THEN MAX = B;
  IF (C > A) AND (C > B) THEN MAX = C;
  IF (A < B) AND (A < C) THEN MIN = A;
  IF (B < A) AND (B < C) THEN MIN = B;
  IF (C < A) AND (C < B) THEN MIN = C;
  writeln ('МАКСИМАЛЬНОЕ ЧИСЛО ', MAX);
  writeln ('МИНИМАЛЬНОЕ ЧИСЛО ', MIN);
  Readln;
END.

```

7. Расчет функции

$$y = \begin{cases} x+3, & \text{если } x < 0 \\ x^2, & \text{если } 0 \leq x \leq 4 \\ e^{x-6}, & \text{если } x > 4 \end{cases}$$

Записать программу в тетрадь.

```
Program yslovie;  
Var x, y: real;  
Begin  
  writeln ('x=');  
  Readln (x);  
  If x<0 then y:=x+3  
  else  
  If x>4 then y:=exp(x-6)  
  else y:=sqr(x);  
  writeln ('y',x:8:2);  
  Readln;  
End.
```

8. С использованием оператор case составить программу для вычисления функции  $Y = (1 + X + X^2)^N$ , в которой N принимает целые значения (1, 2, 3). Записать программу в тетрадь.

```
Program Degree;  
var  
N: integer;  
X, Y: real;  
begin  
  writeln ('Введите значение основания X');  
  Readln (X) ;  
  writeln ('Введите показатель степени от 1 до 3') ;  
  Readln (N) ;  
  X: =1+X+X*X;  
  case N of {Вычисление значения селектора и выбор варианта}  
  1 :  writeln ('Y=',X) ;  
  2 :  writeln ('Y=', X*X) ;  
  3 :  writeln ('Y=', X*X*X)  
  else  
    writeln ('Нет данных');  
  end;  
  readln;  
end.
```

В этом примере на экран выводится запрос на ввод значения аргумента X и показателя степени N.

Введенный показатель используется затем как значение селектора.

В зависимости от значения селектора реализуется тот или иной вариант расчета функции X1:

- если N=1 - то вариант, помеченный как 1,

- если N=2 - то вариант с меткой 2 и т.д.
  - Если в качестве N в начале работы программы будет введено число, не значащееся среди констант выбора (например, 9), то на экран будет выведено сообщение «Нет данных».
9. С использованием оператор case составить программу случайного предсказания одного из десяти вариантов ближайшего будущего с вероятностью 1/20, в остальных случаях - вы "неудачник". Записать программу в тетрадь.

```

PROGRAM FUTURE;
var N: word;
Begin
  writeln('ПРЕДСКАЗАНИЕ БУДУЩЕГО');
  Randomize; N:=Random(20)+1; { N - случайное число от 1 до
20 }
      writeln;
write('Вас ожидает - ');
  writeln ('Введите случайное число N от 1 до 20 ');
Readln (N);
  case N of
    1:  writeln('счастье');
    2:  writeln('пятерка');
    3:  writeln('дорога');
    4:  writeln('двойка');
    5:  writeln('болезнь');
    6:  writeln('здоровье');
    7:  writeln('деньги');
    8:  writeln('любовь');
    9:  writeln('встреча');
      10:  writeln('дети')
  else
      writeln('неудача')
  end;
      writeln('Нажми Enter');
readln;
END.

```

10. Составим программу, с помощью которой можно было бы ввести номер месяца, а программа ответила бы, какому времени года соответствует этот месяц. Записать программу в тетрадь.

Приведем простой пример селектора, содержащего списки констант выбора.

Составим программу, с помощью которой можно было бы ввести номер месяца, а программа ответила бы, какому времени года соответствует этот месяц.

```

Program Season;
Var
  N: integer;

Begin
  writeln ('Введите номер месяца');
  Readln (N);
  Case N of

```

```

1, 2, 12: writeln ('Это зима');
3..5:    writeln ('Это весна');
6..8:    writeln ('Это лето');
9..11:   writeln ('Это осень');
Else
  writeln ('Нет такого месяца');
End;
Readln;
End.

```

11. Составим программу для определения порядка целого числа N от 0 до 999. Записать программу в тетрадь.

```

      case N of
        0..9:  writeln('однозначное');
        10..99: writeln('двузначное');
        100..999: writeln('трехзначное')
      else writeln('Число "N" не входит в указанный диа-
пазон')
Readln
end;

```

12. Вводится целое число, если это цифра, то определить четная она или нет, а если число, то определить попадает ли оно в диапазон от 10 до 100, если нет, то выдать соответствующее сообщение. Записать программу в тетрадь.

```

program chislo;
var i:integer;
begin
write('Введите целое число: ');
readln(i);
case i of
0,2,4,6,8 : writeln('Четная цифра');
1,3,5,7,9 : writeln('Нечетная цифра');
10..100,200: writeln('Число от 10 до 100 или 200');
else writeln('Число либо отрицательное, либо > 100, но не
200');
end;
readln
end.

```

13. Задайте с помощью команды выбор вычисления по формулам. Записать программу в тетрадь.

$$z = \begin{cases} 1 & , \text{ если } c=0, \\ x & , \text{ если } c=1, \\ 3x^2 - 1/2 & , \text{ если } c=2, \\ x^3 - 3x/2 & , \text{ если } c=3, \\ 2x^4 - 3x/2 & \text{ в противном} \\ & \text{случае} \end{cases}$$

## Практическая работа №23 Оператор While

### Цель:

- 1) Освоить простейшую структуру программы с использованием операторов цикла.
- 2) Получить практические навыки работы над ошибками.

### Теоретический материал

Этот оператор является оператором цикла с предусловием («цикл пока») – т. к. проверка условия цикла производится при каждом повторении перед выполнением тела цикла.

В цикле While... "оператор" выполняется если условие верно, если условие ложно, то цикл заканчивается, т. е. цикл While... повторяется пока выполняется условие.

Формат записи:

while <условие> do <тело цикла> (пока выполняется условие, делать).

While, Do – пока выполняется условие, делать.

### ХОД РАБОТЫ:

1. С клавиатуры вводятся два натуральных числа: А и В. Требуется найти наибольший общий делитель (НОД) по алгоритму Евклида (Пока числа не равны друг другу, производится их сравнение и из большего числа вычитается меньшее; при равенстве чисел за результат принимается любое из них). Записать программу в тетрадь.

```

Program example;
Var A, B: Integer;
begin
  writeln ('введите два натуральных числа');
  Read (A, B);
  While A<>B do
    If A>B then A:=A-B else B:= B-A;
    writeln (A:10); {вывод результата}
  Readln;
end.

```

2. Задача табулирования (вычисления значений) функции. Дана функция  $Y = \frac{1}{X}$ , где X изменяется от А до В с шагом Н. Вычислить значения функции Y с учетом ее области допустимых значений (ОДЗ):  $X \neq 0$  (X не равно 0). Записать программу в тетрадь.

```

Program example;
Var A, B, H, X, Y: REAL;
Begin
  writeln ('введите границы отрезка и шаг H');
  Readln (A, B, H);
  X:=A;
  While X<B+H/2 do
  begin
    If X<>0 then begin Y:=1/X;
      writeln ('При X=',X:8:2,' Y=', Y:8:2);{вывод результата}
    end
    else writeln ('При X=',X:8:2,' значение Y не вычисляется');{вывод результата}
    X:=X+H;
  end;
  Readln;
End.

```

3. Дана функция  $Y = 3X^3 + 1,5X^2 - 1,8$ , где X изменяется от A до B с шагом H. Найти максимальное значение Y на отрезке [A,B]. Записать программу в тетрадь.

```

Program example;
Var A, B, H, X, Y, MAX: REAL;
begin
  writeln ('введите границы отрезка и шаг H');
  Read (A, B, H);
  X:=A;
  MAX:=3*SQR(X)*X+1.5*SQR(X)-1.8;
  While X<B+H/2 do
  Begin
    Y:=3*SQR(X)*X+1.5*SQR(X)-1.8;
    If Y>MAX then MAX:=Y;
    X:= X+H;
  End;
  writeln ('Максимум=',MAX:8:3);{вывод результата}
  Readln;
end.

```

4. В первый день пловец проплыл 3 км. В каждый последующий день он проплывал на 10 % больше, чем в предыдущий.
- В какой по счету день пловец начнет проплывать более 5 км?
  - К какому дню он суммарно проплывет более 30 км?
- Записать программу в тетрадь.

Решение:

Введем обозначения:

D – номер дня;

N – расстояние, проплываемое в очередной день;

S – суммарное расстояние;



В какой по счету день пловец начнет проплывать более 5 км?

```
Program example;  
Var N: REAL;  
    D: integer;  
Begin  
N:=3;  
D:=1;  
While N<=5 do begin  
N:=N+0.1*N;  
D:= D+1;  
End;  
  writeln ('Номер дня',D:6);{вывод результата}  
Readln;  
end.
```

К какому дню он суммарно проплывет более 30 км?

```
Program example;  
Var N, S: REAL;  
    D: integer;  
Begin  
N:=3;  
S:=3;  
  
D:=1;  
While N<=30 do begin  
N:=N+0.1*N;  
S:= S+1;  
D:= D+1;  
End;  
  writeln ('Номер дня',D:6);{вывод результата}  
Readln;  
end.
```

5. Дано натуральное число n. Подсчитать количество цифр данного числа. Записать программу в тетрадь.

```
Program example;  
Var m, n: integer;  
k: integer; {счетчик цифр}  
begin  
  writeln ('введите целое число');  
Readln (n);  
k:=0;  
m:=n  
While m<>0 do  
begin  
k:=k+1; {увеличиваем счетчик на единицу}  
m:=m div 10; {уменьшаем число на последнюю цифру}  
end;
```

```

    writeln ('количество цифр в числе ', n, ' равно ', k); {вывод
результата}
Readln;
end.

```

6. Составим программу для вычисления суммы ряда  $S = X^1 + X^2 + \dots + X^N$  при произвольном значении  $X$ . Записать программу в тетрадь.

```

Program Sum;
var
I, N: integer;
X, Y, S: real;
begin
    writeln ('Введите X, N');
Readln (X, N) ;
I:=1;
Y:=1;
S:=0;
while I<=N do
begin
Y:=Y*X;
S:=S+Y;
I:=I+1;
end;
    writeln ('Сумма= ', S);
Readln;
End.

```

## Практическая работа №24 Оператор Repeat Until

### Цель:

- 1) Освоить простейшую структуру программы с использованием оператора цикла Repeat...Until.
- 2) Получить практические навыки работы над ошибками.

### Теоретический материал

#### Пример цикла типа repeat ... until

Определить число  $n$ , при котором сумма квадратов натурального ряда чисел от 1 до  $n$  не превысит величину  $K$ , введенную с клавиатуры. Т.е.

$$S \geq K, \text{ где } S =$$

```

program sum_sq; {Сумма квадратов натурального ряда }
uses crt;
var
k, s, n : integer;
begin
clrscr ;
    writeln( 'Введите K' );
readln (k);

```

```

s:=0;    n:=1;
repeat
  s :=s+n*n;
  n := n+1;
until s > k;
  writeln ('N= ', n : 3, '   s= ' , s : 5 );
readln;
end.

```

Цикл повторяется до тех пор, пока условие записанное после ключевого слова until, будет ложным (не выполняется). Как только это условие выполнится, происходит выход из цикла. После окончания цикла производится печать результата (оператор writeln). Отметим, что цикл с предусловием (типа while) может не выполниться ни разу, цикл с постусловием repeat ... until выполнится по крайней мере 1 раз. Когда число повторений цикла неизвестно заранее, применяются циклы с предусловием или с постусловием. Когда число повторений цикла известно заранее, как правило, применяется цикл типа for. Но любой цикл типа for можно заменить циклом с предусловием или постусловием.

### ХОД РАБОТЫ:

1. Составим программу, с помощью которой можно было бы проверить формулу убывающей геометрической прогрессии ( $q < 1$ ):  $j = 1, q, q^2, \dots, q^N$  на конкретных примерах. Записать программу в тетрадь.

1) Из математики, возможно, вы знаете, что сумма членов этой бесконечной прогрессии равна  $1/(1-q)$ .

2) Программа, как известно, не в состоянии вычислить сумму бесконечного числа членов прогрессии, даже если прогрессия и убывающая.

3) Поэтому мы построим алгоритм таким образом, чтобы вычисление суммы прекращалось, как только последний член в этой сумме будет меньше некоторой заранее заданной величины  $E$  (например,  $E=0.0001$ ).

4) В следующей программе выполняется ввод значений  $q$ , и суммирование членов прогрессии с помощью оператора repeat:

```

program Progression;
var I, N: integer;
    Q, Y, S, E: real;
begin
  writeln ('Введите Q');
  Readln (Q);
  writeln ('Введите границу для последнего члена прогрессии');
  Readln (E);
  Y :=1;
  S:=1;
  Repeat
  Begin
  Y:=Y*Q;
  S:=S+Y ;
  End ;
  Until Abs (Y) <=E;

```

```

    writeln ('Сумма=', S);
Readln
End.

```

- 2 Выводится на экран последовательность 20 случайных чисел из промежутка [-10,10]. Определить количество отрицательных чисел больших -5 в данной последовательности

```

program psldv;
var
i,j:integer;
x:real;
begin
randomize; {процедура инициализации генератора случайных чисел}
i:=20; j:=0;
repeat
x:=random(20)-10; {генерируем случайное число}
write(x:4:0); {выводим это число на экран}
If (x > -5) and (x < 0) then
inc(j); {процедурой inc увеличиваем аргумент на единицу, считаем найденные числа}
dec(i); {процедурой dec уменьшаем аргумент на единицу, определяем условие выхода из цикла}
until i=0;
writeln;
writeln ('Количество отрицательных чисел больших -5 равно ', j);
end.

```

## Практическая работа №25 Операторы цикла For. Вложенные циклы

### Цель:

- 1) Освоить простейшую структуру программы с использованием оператора цикла FOR.
- 2) Получить практические навыки работы над ошибками.

### Теоретический материал

В цикле типа for число повторений известно заранее, поэтому применяется при выполнении расчетов, повторяющихся определенное количество раз. Оператор имеет вид:

```
For <параметр>:=<нач. знач.> To <кон. знач.> Do <оператор>;
```

Например:

```
For i:= N1 To N2 Do "оператор";
```

i - параметр цикла,

N1, N2 - начальное и конечное значения параметра цикла

Если требуется выполнить в цикле несколько операторов, они заключаются в операторные скобки begin ... end, образуя составной оператор:

```

For i:= N1 To N2 Do
Begin
"оператор1";
"оператор2";
...
End;

```

С помощью цикла типа for удобно находить суммы, произведения, искать максимальные и минимальные значения и т.п. При нахождении суммы некоторой переменной, например S присваивается значение 0, затем в цикле к этой переменной прибавляется соответствующий член заданной последовательности. При нахождении произведения переменной присваивается значение 1, затем в цикле эта переменная умножается на общий член последовательности.

### ХОД РАБОТЫ:

1. Ввести с клавиатуры произвольное целое число N и вычислить сумму всех целых чисел от 1 до N. Записать программу в тетрадь.

```

Var i, n, s: integer;
begin
  writeln ('N=');
  Readln (n); {ввести N}
  s:=0; {начальное значение суммы}
  for i:=1 to n do s:=s+i; {цикл подсчета суммы}
  writeln ('сумма=', s);
  Readln
end.

```

2. Разработать программу расчета суммы 12 объемов.  
 $V = (4 * \pi * R * R * R) / 3$ . Записать программу в тетрадь

```

Program pr25;
Uses CRT;
Const
Pi=3.14; h=0.05;
Var i: integer;
R, V: real;
begin
CLRSCR;
V:=0; R:=9.95;
for i:=1 to 12 do
begin
  V:=V+(4*Pi*R*R*R)/3;
  R:=R+h;
end;
  writeln ('объем = ', V);
readln;
end.

```

3. Разработать программу расчета функции. Записать программу в тетрадь.

$$y = \sum_{i=1}^n (i^2) = 1^2 + 2^2 + \dots + n^2$$

Исходные данные: n – количество слагаемых;

Входные данные: y – значение функции.

```
Program example;
Uses CRT;
Var n, i, y, SL: integer;
begin
CLRSCR;
Write ('введите n=>');
Readln (n);
y:=0;
for i:=1 to n do
begin
SL:=sqr(i);
y:=y+SL;
end;
write ('y:=', y);
readln;
end.
```

4. Найти сумму чисел обратных квадратам 100 первых нат-х чисел. Записать программу в тетрадь.

```
program jj;
var i:integer;
s,x:real;
begin
s:=0;
for i:=1 to 100 do
begin
x:=1/(i*i);
s:=s+x;
end;
writeln('s=', s:5:10);
readln;
end.
```

Оператор цикла часто применяется для суммирования значений некоторой последовательности чисел или значений функции при известном числе операций суммирования. Напомним некоторые определения, связанные с расчетом суммы последовательности.

Сумма членов последовательности величин  $a_1, a_2, a_3, \dots, a_N$  называется конечной суммой  $S_N = a_1 + a_2 + a_3 + \dots + a_N$ . Для некоторых последовательностей известны формулы расчета конечных сумм, например:

при  $a_N = a_{N-1} + d$ ;  $S_N = (a_1 + a_N) * N / 2$ ; - арифметическая прогрессия,

при  $a_N = a_{N-1} * q$ ;  $S_N = (a_1 - a_N * q) / (1 - q)$ ; - геометрическая прогрессия,

где  $d$  и  $q$  - постоянные числа.

Здесь  $N$ -ый член последовательности выражается через  $(N-1)$ -ый член. Такие зависимости называются рекуррентными.

Конечная сумма последовательности может быть неизвестна, тогда для ее расчета применяется алгоритм суммирования членов последовательности в цикле от 1 до  $N$ .

Приведем пример расчета конечной суммы последовательности:

$$1^2 + 3^2 + 5^2 + \dots + (2*N-1)^2; \quad S_N = N*(4*N^2-1)/3;$$

5. Составим программу для расчета конечной суммы последовательности  $1^2 + 3^2 + 5^2 + \dots + (2*N-1)^2$ ;  $S_N = N*(4*N^2-1)/3$ ;

Записать программу в тетрадь

```
PROGRAM SUM_K; { расчет конечной суммы }
  var a, S, Sn, i, N: word;
Begin
  writeln('Введите число членов суммы N=');
  readln(N);
  S:= 0;
  For i:= 1 to N do
    begin      { цикл суммирования }
      a:= Sqr(2*i-1);
      S:= S+a
    end;
  Sn:= N*(4*N*N-1) div 3;

  writeln('Конечная сумма S=', S:10);
  writeln('Расчет конечной суммы по формуле Sn=',
Sn:10);
  writeln('Нажми Enter');
Readln;
End.
```

6. В некоторых случаях  $N$ -ый член последовательности определяется через сумму предыдущих членов, например,  $a_N = p*S_{N-1}$ , тогда  $S_N = S_{N-1} + a_N = S_{N-1}*(1+p)$ , и конечную сумму можно рассчитать по формуле:  $S_N = S_0*(1+p)^N$ , где " $S_0$ " - начальная сумма.

Рассмотрим программу вычисления конечной суммы денежного вклада в банк через  $N$  месяцев при ежемесячной процентной ставке " $pr$ " (5% соответствует  $pr=5$ ). Записать программу в тетрадь.

```
PROGRAM VKLAD; { расчет конечной суммы вклада в банк }
  var S, Sn, pr: real;
      i, N: integer;
Begin
  writeln('Введите начальную сумму вклада S=');
  readln(S);
  writeln('Введите процент по вкладу pr=');
  readln(pr);
  writeln('Введите количество месяцев вклада N=');
  readln(N);
```

```

        For i:= 1 to N do S:= S*(1+pr/100); { цикл произведе-
ний }

        writeln('Конечная сумма вклада S=', S:10:2);

        {Оператор для расчета "Sn" напишите самостоятельно
}

    writeln('Расчет конечной суммы вклада по формуле Sn=',
Sn:10:2);
    writeln('Нажмите Enter');
    readln
End.

```

## Практическая работа №26 Массивы. Операции с массивами. Одномерные массивы

### Цель:

- 1) Освоить простейшую структуру программы при работе с массивами чисел.
- 2) Получить практические навыки работы в интерактивном (диалоговом) режиме.
- 3) Получить практические навыки работы над ошибками.

### Теоретический материал

Массивы - структурированный тип данных с элементами одного типа. Количество элементов определяет размер массива. Например, массив составляют заработные платы сотрудников подразделения предприятия, здесь число элементов равно числу сотрудников; массив образуют набор чисел, их количество равно числу элементов массива. Номер элемента массива называется его индексом. Массив может иметь не один, а большее число индексов. Число индексов называется размерностью массива, например, массив с двумя индексами называется двумерным массивом. Таким двумерным массивом является, в частности, матрица системы  $n$  линейных алгебраических уравнений с  $n$  неизвестными. В то же время столбец свободных членов этой системы является одномерным массивом.

Массив должен быть описан либо в разделе переменных VAR, либо в разделе типов TYPE следующим образом:

```

TYPE <имя_типа> = ARRAY [t1,t2,...,tn] OF <тип_элементов_массива>;
VAR <имя_массива> : <имя_типа>;

```

или

```

VAR<имя_массива>: ARRAY [t1,t2,...,tn] OF<тип_элементов_массива>;

```

где  $t_1, \dots, t_n$  - тип индекса (перечислимый или интервальный);

Например:

```

VAR a:ARRAY[1..5] OF real;

```

Описан массив  $a$  действительных чисел, который состоит из 5 элементов.

```

TYPE t = ARRAY[1..3,1..4] OF integer;

```

```

VAR b: t;

```

Описана матрица  $b$  из 3 строк и 4 столбцов, элементы которой являются целыми числами.



```

TYPE
  t1 = ARRAY [1..4] OF integer;
  t = ARRAY [1..5] OF t1;
VAR   d: t;
       k: t1;

```

В начале описан тип одной строки t1, затем тип всей матрицы t через тип строки t1. В разделе переменных указано, что d - двумерный массив размером (5,4), а k - одномерный массив размером (4).

### Пример обработки одномерного массива

Дан одномерный массив MAS(12) из вещественных чисел. Найти наибольший элемент массива и его индекс.

```

program pr4_1 ;
const  n = 12;    {константа n определяет размер массива в
описании}
type
m = array [1..n] of  real ;      {m - тип массива mas  }
var    mas : m ;
i , num : integer;
max : real;
begin
for i:=1 to n do          {ввод элементов массива mas по 1 в
строке}
begin
  writeln('введите элемент массива', i) ;
  read  (mas[i]);
end;
num :=1;  max := mas [1];
for i: =2 to n do
if mas [ i ] > max then    begin
max := mas [i ]; num := i ;
end;
  writeln;      {вывод массива в строку}
for i:=1 to n do  write (mas[i]:5:1);    writeln;
  writeln ('максимальный элемент = ', max:4:1, ' его ин-
декс=' , num );
readln;
end.

```

Переменная max сравнивается с элементами массива, и если элемент массива больше max, то переменной max присваивается значение элемента массива, а переменной num - индекс этого элемента. По окончании цикла переменная max будет иметь значение, равное максимальному элементу массива, а переменная num - значение индекса этого элемента.

ХОД РАБОТЫ:

1. Дан массив [1..10]. Посчитать количество элементов больших 6 и количество элементов меньших и равных 6. Записать программу в тетрадь.

```
Program as;  
Uses Crt;  
VAR A:ARRAY[1..10] OF integer;  
      S, D, I: INTEGER;  
BEGIN  
ClrScr;  
  writeln('Введите элементы таблицы:');  
  FOR I:=1 TO 10 DO READLN(A[I]);  
  S:=0;  
  D:=0;  
  FOR I:=1 TO 10 DO  
  IF A[I]>6 THEN S:=S+1 else D:=D+1;  
    WRITELN('Количество элементов больших 6: ',S) ;  
    WRITELN('Количество элементов меньших и равных 6: ',D);  
  readln;  
END.
```

2. Дан массив [1..5]. Найти минимальный элемент массива. Записать программу в тетрадь.

```
Program be;  
Uses CRT;  
  var i, min: integer;  
      a: array[1..5] of integer;  
BEGIN  
  ClrScr;  
  writeln('Введите элементы таблицы');  
  for i:=1 to 5 do readln(a[i]);  
  min:=a[1];  
  for i:=1 to 5 do If a[i]<min then min:=a[i];  
    writeln('min=',min);  
  readln;  
END.
```

3. Дан массив [1..7]. Найти максимальный элемент массива. Записать программу в тетрадь.

```
Program we;  
Uses CRT;  
  var i, max: integer;  
      a: array[1..7] of integer;  
BEGIN  
  ClrScr;  
  writeln('Введите элементы таблицы');  
  for i:=1 to 7 do readln(a[i]);  
  max:=a[1];  
  for i:=1 to 7 do If a[i]>max then max:=a[i];
```

```

    writeln('max=',max);
  readln;
END.

```

4. Подсчитать количество отрицательных и положительных чисел в таблице. Записать программу в тетрадь.

```

program ff;
var A:array[1..10] of integer;
    I : integer;
    z1, z2: integer;
begin
  writeln('введите таблицу A[1..10]');
  for i:=1 to 10 do
    readln(a[i]);
    z1:=0;z2:=0;
  for i:=1 to 10 do
    begin
      If a[i]>0 then      z1:=z1+1;
      If a[i]<0 then z2:=z2+1;
    end;
    writeln('z1=',z1,' z2=',z2);
  readln;
end.

```

5. Определить, какая задача решается при выполнении данной программы. Записать программу в тетрадь.

```

program qw;
var a: array[1..10] of integer;
    i: integer;
begin
  writeln('введите массив');
  for i:=1 to 10 do
    readln(a[i]);
  for i:=1 to 10 do
    If a[i]>0 then a[i]:=-1;
  for i:=1 to 10 do
    writeln(a[i]);
  readln;
end.

```

6. В данном тексте заменить 'b' на другую букву. Записать программу в тетрадь.

```

program qq;
uses crt;
VAR st: string;
    i: integer;
BEGIN      clrscr;

```

```

    writeln('введите текст');
    readln(st);
    for i:=1 to length(st) do
        If st[i]='B' then st[i]:='A';
    writeln(st);
readln;
END.

```

## Практическая работа №27 Массивы. Операции с массивами. Двумерные массивы

### Цель:

- 1) Освоить простейшую структуру программы при работе с массивами чисел.
- 2) Получить практические навыки работы в интерактивном (диалоговом) режиме.
- 3) Получить практические навыки работы над ошибками.

### Теоретический материал

#### Пример обработки двумерного массива

Как правило, при обработке многомерных массивов используются вложенные циклы, т.е. цикл по столбцам располагается внутри цикла по строкам.

Дана матрица  $A(3,4)$ , и вектор  $B(4)$ , состоящие из целых чисел. Умножить матрицу  $A$  на вектор  $B$ .

```

program pr4-2 ;
const m=3;    n=4;
var
a  : array [ 1 .. m, 1 .. n ] of    integer; {описание матрицы }
b   : array [ 1 .. n ] of    integer;
{описание вектора }
c  : array [ 1 .. m ] of integer;           {
описание C }
i, j: integer;
begin
for i:=1 to m do                {ввод матрицы }
begin
    writeln ('введите элементы ', i , '-той строки');
    for j:=1 to n do read (a [i, j] ); writeln;
end;
    writeln ('введите элементы вектора');
for j:=1 to n do                { ввод вектора }
read (b[ j]);    writeln;
for i:=1 to m do
begin
c [ i ]:=0;        for j:=1 to n do        c[i] := c[ i ]+
a[i , j]* b[j];
end;

```

```

for i:=1 to m do      {форматный вывод матрицы }
begin
for j:=1 to n do write (a [i, j]: 4 );   writeln;
end;
for j:=1 to n do write (b [ j ] :4);      {вывод массива B }
  writeln ;
for i:=1 to m do write (c [ i ] :4);      { вывод массива C
}
readln;
end.

```

В программе элементы матрицы вводятся по строкам по одному с подтверждением клавишей Enter. А выводятся в общепринятом виде: каждая строка матрицы с новой строки экрана (цикл i по строкам внешний, а цикл j – внутренний)

#### ХОД РАБОТЫ:

1. Определить какая задача решается при выполнении данной программы. Записать программу в тетрадь.

```

program D_57;
  uses CRT;
  var t, s: string;
      n:integer;
BEGIN
  ClrScr;
  Write('Введите текст:');
readln(t);
  write('Введите слово:');
readln(s);
  n:=pos(s,t);
  If n=0 then writeln('текст не содержит этого слова')
    else writeln('текст содержит слово');
readln;
  END.

```

2. Определить какая задача решается при выполнении данной программы. Записать программу в тетрадь.

```

program A;
  uses CRT;
  label m;
  var st, sq: string;
      i, j: integer;
BEGIN
  ClrScr;
  write('Введите текст:'); readln(st);
  sq:='';
  for i:=1 to length(st) do
    begin

```

```

m:If st[i]='c' then
    begin
        sq:=sq+'aa';
        i:=i+1;
        If st[i]='c' then
            begin
                sq:=sq+'aa';
                i:=i+1;
                goto m
            end;
        end;
        sq:=sq+st[i];
    end;
writeln('получили ',sq);
readln;
END.

```

3. Определить какая задача решается при выполнении данной программы. Записать программу в тетрадь.

```

program Z;
uses CRT;
var st, b: string;
    n: integer;
BEGIN
ClrScr;
write('Введите текст:');readln(st);
writeln('введите сочитание букв');
readln(b);
repeat
n:=pos(b,st);
If n<>0 then
    begin
        delete(st,n,2);
        insert('m',st,n);
    end;
until(n=0);
writeln(st);
readln;
END.

```

4. Проверить, есть ли в данном тексте буква 'б', если есть, то заменить последнюю из них на букву 'а'. Записать программу в тетрадь.

```

program ZAD_62;
uses CRT;
var a, st: string[100];
    k, i: integer;

```

{буквы русские}

```

BEGIN
  ClrScr;
  a:='В тексте нет буквы <б>';
  write('Введите текст: '); readln(st);
  for i:=1 to length(st) do
    If st[i]='б' then
      begin
        a:='В тексте есть буква <б>';
        k:=i;
      end;
  If a='В тексте есть буква <б>' then
    begin
      delete(st,k,1);
      insert('a',st,k);
    end;
    writeln(a);
    writeln('Результат:',st);
  readln;
END.

```

#### 5. Нахождение суммы элементов массива

Пусть известно, что в автопарке, имеющем 18 машин марки КАМАЗ, каждый из КАМАЗов перевез за день определенный объем груза. Определить суммарный объем перевозок грузов за день.

При решении задачи будем использовать тип массива КАМАЗ для описания всех КАМАЗов автопарка; переменную P[i] для описания объема груза, перевезенной i-ой машиной за день (i меняется от 1 до 18).

Текст программы может иметь при этом следующий вид:

```

Program pr1;
Type КАМАЗ =array[1..18] of real;
Var i:integer;
      P:КАМАЗ;
      S:real;

Begin
  S:=0;
  For i:=1 to 18 do
    Begin
      writeln ('Введите объем перевозок ', i, '-й машины, т');
      Readln (p[i]);
      S:=S+p[i];
    End;
    writeln ('Суммарный объем перевозок S=',S:8:2,' т');
  Readln
End.

```

#### 6. Нахождение наибольшего элемента в массиве

Накопление суммы в данном примере будет проводиться по шагам, при вводе для очередной машины значения объема перевозок сумма будет увеличиваться на данную величину.

Аналогично реализуется и алгоритм нахождения произведения элементов массива (с заменой начального значения суммы  $S:=0$  на начальное значение произведения  $S:=1$ , и с заменой операции сложения элементов массива «+» на операцию умножения «\*»).

Известна среднемесячная зарплата всех 16 сотрудников одного отдела. Найти величину наибольшей среднемесячной зарплаты в отделе.

Для описания списка зарплат сотрудников воспользуемся типом массива ZARPL, для задания зарплаты каждого из сотрудников – переменной SOTR.

Текст программы к приведенному примеру:

```
Program pr2;  
Type ZARPL=array[1..16] of real;  
Var i:integer;  
      sotr:ZARPL;  
      Max: real;  
Begin  
For i:=1 to 16 do  
Begin  
  writeln ('Введите зарплату ', i, ' -го сотрудника, руб. ');  
  Readln(sotr[i]);  
End;  
Max:=sotr[1];  
For i:=2 to 16 do If max<sotr[i] then max:=sotr[i];  
  writeln ('Наибольшая зарплата =', max:10:2, ' руб. ');  
Readln  
End.
```

В данной программе реализован следующий алгоритм. В начальный момент предполагается, что наибольшей является зарплата первого из сотрудников. Далее величина этой зарплаты сравнивается с зарплатами остальных сотрудников, и, если найдется зарплата, большая, чем предполагаемый максимум, то переменной Max присваивается это значение.

Аналогично может быть найден и наименьший элемент в массиве. Для этого достаточно заменить в изложенной программе условие  $\text{max} < \text{sotr}[i]$  на условие  $\text{min} > \text{sotr}[i]$ , где  $i$  меняется от 1 до 1 до 16,  $\text{min}$  – переменная типа REAL.

7. Нахождение количества элементов массива, удовлетворяющих некоторому условию. Известны результаты экзамена 22 студентов одной группы по информатике. Определить, сколько студентов сдали экзамен на 4 и 5.

Один из вариантов решения поставленной задачи имеет следующий вид:

```
Program pr3;  
Label 1;  
Type INF=array[1..22] of integer;  
Var stud: INF;  
      i, p: integer;  
begin  
p:=0;  
for i:=1 to 22 do  
begin
```



```

1:   writeln ('Введите оценку ',i,'-го студента');
Readln (stud[i]);
If (stud[i]<1) or (stud[i]>5) then goto 1;
If stud[i]>3 then p:=p+1;
end;
   writeln('На 4 и 5 сдали экзамен ',p:2,' студентов');
Readln
end.

```

В данной программе для обозначения списка оценок по информатике использовался тип массива INF, для обозначения оценок конкретных студентов – переменная stud. Программа предусматривает проверку корректности вводимых данных: при попытке ввода несуществующей по пятибалльной системе оценки, программа повторяет ее ввод. Для этого используется оператор перехода GOTO, где имя метки, к которой осуществляется переход (в данном случае 1), описывается в разделе описания меток LABEL.

8. Нахождение элементов массива, удовлетворяющих определенному условию:  
Известны результаты 5 студентов по итогам экзаменов по химии и информатике.  
Найти фамилии студентов, сдавших оба экзамена на отлично.

Для решения поставленной задачи может быть использована следующая программа:

```

program Sessia;
type PR=array [1..5,1..2]of integer;
      Fam=array[1..5]of string[10];
var   r :pr;
      st: fam;
      i, j: integer;
begin
for i:=1 to 5 do
begin
   writeln('Введите фамилию ',i,'-го студента ');
readln(st[i]);
   writeln('Введите оценку данного студента по химии (от 2 до 5) ');
readln(r[i,1]);
   writeln('Введите оценку данного студента по информатике (от 2 до 5) ');
readln(r[i,2]);
end;
for i:=1 to 5 do
If (r[i,1]=5) and (r[i,2]=5) then
   writeln('Студент-отличник - ',st[i]);
end.

```

В данной программе для хранения фамилий студентов используется одномерный строковый массив st типа Fam, для хранения оценок студентов – двумерный целочисленный массив r типа PR, причем первый столбец матрицы r используется для хранения результатов экзамена по химии, второй столбец – экзамена по информатике. Если у некото-

рого студента оценки за оба экзамена составили 5 баллов, то его фамилия будет выведена на экран с сообщением «Студент-отличник».

#### 9. Нахождение сумм элементов строк матрицы.

Рассмотрим задачу нахождения сумм элементов строк матрицы на примере задачи подсчета итогов футбольного чемпионата.

Пусть задана таблица результатов игр 5 команд футбольного чемпионата размера 5x5. На диагонали таблицы стоят значения 0, другие элементы таблицы равны 0, 1 или 2, где 0 баллов соответствует проигрышу команды в игре, 1 балл – ничьей, 2 балла – выигрышу. Определить сумму баллов каждой команды по результатам чемпионата.

Легко заметить, что для построения матрицы R результатов игр достаточно ввести лишь стоящую выше (или ниже) главной диагонали половину матрицы, т.к. результаты остальных игр могут быть рассчитаны из известного соотношения: если, например, первая команда обыграла вторую, то элемент  $R[1,2]=2$ , а элемент  $R[2,1]=2-R[1,2]=0$ ; аналогично, если вторая команда сыграла вничью с третьей, то  $R[2,3]=1$ ,  $R[3,2]=2-R[2,3]=1$ . Таким образом, нетрудно получить вид взаимосвязи элементов матрицы:  $R[i,j]+R[j,i]=2$ , где  $i$  и  $j$  меняются от 1 до 5 (кроме элементов главной диагонали). На главной диагонали матрицы R по условию задачи всегда стоят числа 0.

Перейдем к написанию программы решения поставленной задачи.

```
Program foot;
Type tab=array[1..5,1..5] of integer;
Var r: tab;
    i, j, s :integer;
begin
  {ввод стоящих выше диагонали элементов матрицы}
  for i:=1 to 4 do
  for j:=i+1 to 5 do
  begin
    writeln ('Введите результат игры ',i,'-й команды с ',j,' -
й: 0, 1 или 2 балла');
    readln(r[i,j]);
  end;
  {заполнение стоящих на диагонали элементов нулями}
  for i:=1 to 5 do r[i,i]:=0;
  {вычисление стоящих ниже диагонали элементов матрицы}
  for i:=2 to 5 do
  for j:=1 to i-1 do r[i,j]:=2-r[j,i];
  {вывод на экран матрицы результатов игр}
  writeln('Таблица чемпионата');
  for i:=1 to 5 do
  begin
  for j:=1 to 5 do write(r[i,j]:4);
  writeln;
  end;
  {вычисление сумм элементов строк матрицы}
  for i:=1 to 5 do
  begin
  s:=0;
  for j:=1 to 5 do s:=s+r[i,j];
```

```

writeln(i, '-ая команда набрала ', s:3, ' очков');
end;
readln
end.

```

## Практическая работа №28 Символьный тип. Управляющие символы

### Цель:

- 1) Освоить простейшую структуру программы при работе с символьным типом переменных.
- 2) Получить практические навыки работы в интерактивном (диалоговом) режиме.
- 3) Получить практические навыки работы над ошибками.

### Теоретический материал

Понятие строкового типа переменных. Операции над строками.

- 1) Для работы с текстами применяется структурированный тип **string** (строка).
- 2) Строка - это цепочка, составленная из символов.
- 3) Символы берутся из кодовой страницы, поддерживаемой в компьютере.
- 4) Строка похожа на одномерный массив, однако, в отличие от массива, количество элементов строки (символов) не фиксировано.
- 5) Единственное ограничение на количество символов: оно не может превышать 255.
- 6) Строковые переменные должны быть определены в разделе описания переменных.

Пример:

```

var
Text1: string;
T1: string [20];
T2: string [125];

```

В квадратных скобках указывается максимальная длина строки.

- 7) Если длина строки не указана, то принимается длина по умолчанию - 255 символов.

- 8) Можно задать строковый тип.

Пример:

```

type
Atten=string [100];
var
B1: Atten;

```

- 9) Строковые константы определяются текстом, заключенным в кавычки.

Пример:

```

const
Name = 'Константин';

```

- 10) К любому символу строки можно обратиться по его номеру (например, Atten[7]) - аналогично обращению к элементу одномерного массива.

- 11) При этом *i* нужно учесть, что самый первый байт строки имеет номер 0 и содержит значение длины строки.

- 12) Второй байт с номером 1 содержит первый символ строки.

## ХОД РАБОТЫ:

1. Рассмотрим следующую программу. Записать программу в тетрадь.

```
program Hello;      var
    Name  : string[30];
begin
    Write('Как вас зовут? ');
    Readln(Name);
    writeln('Здравствуйте, ',Name);
Readln;
end.
```

Переменная Name описывается с типом string, и для нее отводится пространство под 30 символов. Кроме того, Турбо Паскаль отводит еще один байт, который содержит текущую длину строки. Поэтому независимо от того, какой длины вы вводите имя в ответ на подсказку, по оператору `Writeln` распечатывается имя только указанной длины. Если вы введете имя длиной более 30 символов, то в этом случае только первые 30 символов будут использоваться, а остальные будут проигнорированы.

2. Запишем простейшую программу перевода символов, вводимых с клавиатуры, в числовой код. Записать программу в тетрадь.

```
program SymbolCode;
var
Xsymbol: char;
begin
Write ('Введите символ:');
Readln (Xsymbol);
Writein(Xsymbol,'имеет код',Ord (Xsymbol));
Readln;
end.
```

При вводе символов числовой код может принимать значения от 0 до 255.

### Практическая работа №29 Строковый тип. Операции над строками

#### Цель:

- 1) Освоить простейшую структуру программы при работе со строковым типом переменных.
- 2) Получить практические навыки работы в интерактивном (диалоговом) режиме.
- 3) Получить практические навыки работы над ошибками.

#### Теоретический материал

#### Операции над строками

1. операция **сложения** (конкатенация) - объединение двух слов без пробела.

Пример:

выражение Привет, + 'Игорь' будет давать текст: Привет, Игорь.

2. операции **отношения** (=, <, >, >=, <=) - сравнение двух строк текста.

- Сравнение выполняется посимвольно слева направо до первого несовпадающего символа.

- Большим значением считается то, в котором первый несовпадающий символ имеет больший номер в алфавите.

- Строки считаются равными, если каждый символ одной строки совпадает с каждым символом другой строки в порядке следования символов.

- Результат операции отношения имеет булевский тип, например, выражения Простор'< Просторный' 'depend' >= 'dependence'

имеют соответственно значения True и False.

3. **Присвоение** значения строковой переменной - оператор присваивания:

T1:='Кочеткова Елена';

T2:=T1+'- ученица 8-А класса';

В одном выражении можно записывать переменные как строкового, так и символического типов.

Часто полезными операциями оказываются преобразования данных символического типа char в целое число - код ASCII, а также обратное преобразование.

Преобразование символа в число выполняется функцией Ord, а обратное преобразование - функцией Chr.

№	Процедура или функция	Назначение	Пример	
			Выражение	Результат
1.	Concat (A, B,.. Z)	<u>функция строкового типа</u> , которая возвращает значение сцепленных строк A, B,.. Z. Исходные строки в новой строке следуют в том порядке, в каком они указаны в списке параметров.	Concat (S1, S2, S3)	обитатель лесов кабан
			Concat (S3, S1, S2)	кабан обитатель лесов
2.	Delete (t.poz.n)	<u>процедура удаления N символов в строке T</u> , начиная с позиции под номером Poz.	Delete (S1, 4, 2)	'обитель'
			Delete (S3, 1, 2)	'бан'
3.	Insert (T,S, Poz)	<u>процедура вставки строки T в строку S</u> , начиная с позиции poz.	Insert (S3, S1, 6)	'битакабантель'
4.	Copy (t,Poz,N)	<u>функция строкового типа</u> , которая возвращает значение подстроки длиной N, начиная с позиции Poz в исходной строке T.	Copy (S1, 3, 2)	'ит'
			Copy (S2,2,3)	'аба'
5.	Length (T)	<u>функция целого типа</u> , которая возвращает значение длины строки T.	Length (S1)	9
			Length (S2)	5
			Length (S3)	5

6.	Poz (T1,T2)	<u>функция</u> целого типа, которая обнаруживает в строке T2 первое появление подстроки T1. Функция возвращает номер позиции, в которой находится первый символ подстроки T1. Если подстрока T1 не найдена, то результатом будет 0.	'един' (строка T1) 'объединение' (строка T2).	Выражение Poz (T1, T2) будет иметь результатом число 4.
----	-------------	---	--	---

#### ХОД РАБОТЫ:

1. Составить программу, которая будет выполнять последовательные преобразования слов «холестерин - холерик - болеро». Записать программу в тетрадь.

```

program textconv;
  var
  s1,s2: string;
BEGIN
  s1:='холестерин';
  writeln (s1);
  delete (s1,5,3);
  delete (s1,7,1);
  s2:= s1+'к';
  writeln (s2);
  s2:= copy (s2,2,4);
  s2:= 'б'+s2+'0';
  writeln(s2);
readln;
END.

```

2. Проверить, есть ли в данном тексте буква 'б', если есть, то заменить последнюю из них на букву 'а'. Записать программу в тетрадь.

```

program qwer;
  uses CRT;
  var a, st: string[100]; {буквы русские}
      k, i: integer;
BEGIN
  ClrScr;
  a:='В тексте нет буквы <б>';
  writeln('Введите текст: ');
readln(st);
  for i:=1 to length(st) do
    if st[i]='б' then
      begin
        a:='В тексте есть буква <б>';
        k:=i;
      end;

```

```

If a='В тексте есть буква <б>'
then
    begin
        delete(st,k,1);
    end;
    writeln(a);
    writeln('Результат:',st);
readln;
END.

```

3. Проверить, есть ли в данном тексте сочетание букв. Записать программу в тетрадь.

```

program qwert;
uses CRT;
var st, b: string;
    n: integer;
BEGIN
    ClrScr;
    writeln('Введите текст:');
readln(st);
    writeln('введите соч букв');
readln(b);
    repeat
        n:=pos(b,st);
        If n<>0 then
            begin
                delete(st,n,2);
                insert('m',st,n);
            end;
    until (n=0);
    writeln(st);
readln;
END.

```

4. Нахождение данного слова в тексте. Записать программу в тетрадь.

```

program qwer;
uses CRT;
var t,s:string;
    n:integer;
BEGIN
    ClrScr;
    writeln('Введите текст:');
readln(t);
    writeln('Введите слово:');
readln(s);
    n:=pos(s,t);
    If n=0 then
        writeln('текст не содержит этого слова')

```

```

else writeln('текст содержит слово');
  readln;
end.

```

## Практическая работа №30 Процедуры. Функции

### Цель:

- 1) Освоить простейшую структуру программы при работе с процедурами, функциями.
- 2) Получить практические навыки работы в интерактивном (диалоговом) режиме.
- 3) Получить практические навыки работы над ошибками.

### Теоретический материал

Подпрограмма – это выделенный в самостоятельную единицу участок программы, имеющий собственное имя, которое позволяет вызывать подпрограмму из основной программы.

Два вида подпрограмм:

- процедуры
- функции.

### ХОД РАБОТЫ:

1. Составить программу с использованием процедуры вывода на экран визитной карточки программиста. Записать программу в тетрадь.

```

Program NP_1;
Var Dat, Fam: string;           { Fam: глобальная переменная }
Procedure VIZ(D_R :string); { D_R - формальный параметр }
Var S_t: string;               { S_t: локальная переменная }
Begin
  writeln('| ----- |');
  writeln('|  Разработчик программы:', Fam:14,' |');
  writeln('| ----- |');
  writeln('|  г. Нефтеюганск, ', D_R:14,' |');
  writeln('|  Телефон:  22-44-66 |');
  writeln('| ----- |');
  writeln('|      Комментарий:  ') ;
Readln (S_t);
end;
Begin
  Fam:='И.И.Иванов';
  Dat:='06.12.95';               {Dat - фактический параметр }
  VIZ(Dat);                     { вызов процедуры }
Readln;
END.

```



2. Оформить в виде процедуры программу приветствия. Записать процедуру `Hello` и текст основной программы, из которой выполняются обращения к процедуре. Записать программу в тетрадь.

Эта программа выводит на экран приветствия для трех человек:

1. Кати,
2. Андрея,
3. Лены.

```
Program privet;  
Procedure Hello (Name :string);  
Begin  
    writeln('Привет, ', Name, '!');  
    writeln(Name, ', как дела');  
End;  
Begin  
Hello ('Катя');  
Hello ('Андрей');  
Hello ('Лена');  
Readln;  
End.
```

3. Составим программу вычисления выражений  $f_1 = ch^2(x) + sh(x)$  и  $f_2 = sh^2(x) + ch(x)$  с использованием процедуры расчета гиперболических функций. Записать программу в тетрадь. В некоторых расчетах приходится пользоваться функциями гиперболических синуса и косинуса:

```
Program Hiperbolik;  
Var  
    A: real;  
    Sx, Sy, F1, F2: real;  
Procedure Hpb (X : real; var Ch, Sh : real);  
Begin  
    X:=EXP(X);  
    Ch:=(X+1/X)/2;  
    Sh:=(X-1/X)/2;  
End;  
Begin  
    A:=0.5;  
    Hpb (A, Sx, Sy);  
    F1:=Sx*Sx+Sy;  
    F2:=Sy*Sy+Sx;  
    writeln('Значение функции F1 равно', F1);  
    writeln('Значение функции F2 равно', F2);  
    Readln;  
End.
```

4. Составить программу с использованием функции в расчете выражения  $y = |x+1| + |x+2|$ . Записать программу в тетрадь.

Вычисление модуля оформлено в виде пользовательской функции Modul:

```
Program P1;
Var
  X, Y: real;
Function Modul(x: real): real;
Begin
  If x<0 then Modul:= -x else Modul:=x;
end;
Begin
  writeln('Введите X');
  Readln (X);
  Y:=Modul(X+1)+ Modul(X+2);
  writeln(Y);
  Readln;
End.
```

5. Вычисление степенной функции  $Y=X^N$ , которой нет среди встроенных функций. Для вычисления  $X^N$  предназначена описываемая ниже пользовательская функция. Записать программу в тетрадь.

```
Program Degree;
Var
  Z, F: real;
  N: integer;

Function Deg(N: integer; x: real): real;
Var
  I: integer;
  Y: real;
Begin
  I:=1;
  Y:=1;
  While I<=N do
  Begin
    Y:=Y*X;
    I:=I+1;
  end;
  Deg:=Y;
end;
Begin
  writeln('Введите Z, N');
  Readln (Z, N);
  F:= Deg (N,Z);
  writeln('F=', F);
  Readln;
End.
```

## Практическая работа №31 Создание архива данных. Извлечение данных из архива

**Цель:** Изучение способов и методов архивации данных

### ХОД РАБОТЫ

1. Изучить теоретический материал;
2. Выполнить задания.

### Теоретический материал

**Архивирование данных** - это процесс сжатия файлов, с целью освобождения места на диске.

Часто случается так, что данные не помещаются на дискету или на компакт-диск, а после того как вы выполните архивирование данных, все прекрасно поместится. Особенно хорошо сжимаются тестовые файлы, если повторов очень много, то сжатия можно добиться до 10 раз. Хуже сжимаются цветные графические файлы. Можно сказать, что в среднем архиваторы дают выигрыш в 2-3 раза.

Программа, которая сжимает текстовый файл, называется упаковщиком или архиватором. Программы-упаковщики архивируют не только текстовые файлы, а также программы, звуковые, графические, видеофайлы и другие.

Самораспаковывающийся (SFX, от англ. Self-eXtracting) архив — это архив, к которому присоединен исполнимый модуль. Этот модуль позволяет извлекать файлы простым запуском архива как обычной программы. Таким образом, для извлечения содержимого SFX-архива не требуется дополнительных внешних программ. Тем не менее, WinRAR может работать с SFX-архивом так же, как и с любым другим, поэтому если вы не хотите запускать SFX-архив (например, когда не можете гарантировать, что в нем нет вирусов), то для просмотра или извлечения его содержимого можно использовать WinRAR.

В процессе архивирования данных создается архивный файл, который меньше по объему сжимаемых файлов. После создания архива, сжимаемые файлы можно удалить, тем самым освобождая место на диске.

Если же вам снова понадобилось вернуть архивные файлы в первоначальное состояние, то можно распаковать архив, вернув тем самым файлы на прежнее место. Архив при этом можно удалить, чтобы просто не занимал лишнего места на диске.

Существует достаточное количество архиваторов и столько же типов архивных файлов. Среди них самыми распространенными являются ZIP и RAR.

Если на компьютере никакого архиватора, то можно воспользоваться встроенным архиватором Windows, который отвечает за работу с zip-архивами.

Архивация может использоваться и для создания резервных копий используемых файлов, на случай потери или порчи по каким-либо причинам основной копии (невнимательность пользователя, повреждение магнитного диска, заражение вирусом и т.д.).

Архиваторы позволяют защищать созданные ими архивы паролем, сохранять и восстанавливать структуру подкаталогов, записывать большой архивный файл на несколько дисков (многоотомный архив).

Сжиматься могут как один, так и несколько файлов, которые в сжатом виде помещаются в так называемый архивный файл или архив. Программы большого объема, распространяемые на дискетах, также находятся на них в виде архивов.

**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.

Степень сжатия зависит от используемой программы, метода сжатия и типа исходного файла. Наиболее хорошо сжимаются файлы графических образов, текстовые файлы и файлы данных, для которых степень сжатия может достигать 5 - 40%, меньше сжимаются файлы исполняемых программ и загрузочных модулей — 60 - 90%. Почти не сжимаются архивные файлы. Программы для архивации отличаются используемыми методами сжатия, что соответственно влияет на степень сжатия.

Для того чтобы воспользоваться информацией, запакованной в архив, необходимо архив раскрыть или распаковать. Это делается либо той же программой-архиватором, либо парной к ней программой-разархиватором.

**Разархивация** (распаковка) — процесс восстановления файлов из архива в первоначальном виде. При распаковке файлы извлекаются из архива и помещаются на диск или в оперативную память.

**Самораспаковывающийся архивный файл** — это загрузочный, исполняемый модуль, который способен к самостоятельной разархивации находящихся в нем файлов без использования программы-архиватора.

Самораспаковывающийся архив получил название SFX-архив (Self-eXtracting). Архивы такого типа в обычно создаются в форме .EXE-файла.

Задания:

1. На локальном диске C:, создайте папку «**Архив**», найдите и скопируйте в папку «**Архив**» по два рисунка с расширением \*.jpg и \*.bmp.
2. Сравните размеры файлов \*.bmp и \*.jpg. и запишите данные в таблицу 11.
3. В папку «Архив» поместите файлы \*.doc (не менее 3) и запишите их исходные размеры в таблицу 11.
4. Запустите программу-архиватор **WinZip** «Пуск – Все программы – Zip – Zip File Manager».
5. В появившемся диалоговом окне выберите папку, в которой будет создан архив: C:\Архив. Установите курсор на имя графического файла. Выполните команду «**Добавить (+)**».
6. Введите имя архива в поле Архив – gis\_1.zip и убедитесь, что в поле Формат архива установлен тип Zip.
7. Установите в поле Режим изменения: добавить и заменить.
8. В раскрывающемся списке Уровень сжатия: выберите пункт Нормальный. Запустите процесс архивации кнопкой ОК.
9. Сравните размер исходного файла с размером архивного файла. Данные запишите в таблицу 11.
10. Создайте архив gis\_1.zip, защищенный паролем. Для ввода пароля в диалоговом окне Добавит к архиву в поле Введите пароль: ведите пароль, в поле Повторите пароль: подтвердите пароль. Обратите внимание на флажок Показать пароль. Если он не установлен, пароль при вводе не будет отображаться на экране, а его символы будут заменены подстановочным символом "\*". Это мера защиты пароля от посторонних. Однако в данном случае пользователь не может быть уверен в том, что он набрал пароль правильно. Поэтому при не установленном флажке система запрашивает повторный (контрольный) ввод пароля. Щелкните на кнопке ОК - начнется процесс создания защищенного архива.
11. Выделите gis\_1.zip, выполните команду «**Извлечь**». В появившемся диалоговом окне «**Извлечь**» в поле «**Распаковать в:**» выберите папку – C:\Архив.
12. Щелкните на кнопке ОК. Процесс извлечения данных из архива не запустится, а вместо него откроется диалоговое окно для ввода пароля.

13. Убедитесь в том, что ввод неправильного пароля не позволяет извлечь файлы из архива. Убедитесь в том, что ввод правильного пароля действительно запускает процесс.
14. Удалите созданный вами защищенный архив и извлеченные файлы.
15. Создайте самораспаковывающийся ZIP-архив. Для этого установите курсор на имя архива ris\_1.zip, выполните команду «Добавить (+)».
16. Введите имя архива в поле Архив – rus\_17z и убедитесь, что в поле Формат архива установлен тип 7z.
17. Установите в поле Режим изменения: добавить и заменить, установите флажок «Создать SFX-архив», В раскрывающемся списке Уровень сжатия: выберите пункт Нормальный. Запустите процесс архивации кнопкой ОК.
18. Аналогичным образом создайте архивы для всех файлов в папке «Архив». Сравнительные характеристики исходных файлов и их архивов занести в таблицу 11.

Задание 2.

1. Запустите программу – архиватор **WinRAR** «Пуск – Все программы – WinRAR».
2. В появившемся диалоговом окне выберите папку, в которой будет создан архив: **C:\Архив**.
3. Установите курсор на имя графического файла **ris\_1.jpg**.
4. Выполните команду «Добавить» в появившемся диалоговом окне введите имя архива **ris\_1.rar**. Выберите формат нового архива - **RAR**, метод сжатия - **Обычный?** убедитесь, что в группе «**Параметры архивации**» ни в одном из окошечек нет флажков. Щелкните на кнопке **ОК** для создания архива. Во время архивации отображается окно со статистикой. По окончании архивации окно статистики исчезнет, а созданный архив станет текущим выделенным файлом.
5. Аналогичным образом создайте архивы для файлов в папке «Архив». Сравнительные характеристики исходных файлов и их архивов занести в таблицу 4.
6. Создайте самораспаковывающийся RAR – архив, включающий в себя текстовые и графические файлы.
7. Определите процент сжатия файлов и заполните таблицу 11. Процент сжатия определяется по формуле  $P = \frac{S}{S_0} * 100\%$ , где S – размер архивных файлов, S<sub>0</sub> – размер исходных файлов.

Таблица 4

	Размер исходных файлов	Архиваторы		
		WinZip	WinRar	SFX
<i>Текстовые файлы:</i>				
1. Документ1.doc				
2. Документ2.doc				
3. Документ3.doc				
<i>Графические файлы:</i>				
1. ris_1.jpg				
2. ris_2.bmp				
Процент сжатия <i>текстовой</i> информации (для всех файлов)				
Процент сжатия <i>графической</i> информации (для всех файлов)				

## Практическая работа №32 Использование различных видов АСУ на практике

**Цель:** Изучение способов и методов архивации данных

### ХОД РАБОТЫ

1. Изучить теоретический материал;
2. Создать схему «Виды АСУ»
3. Заполнить таблицу 5.

### Теоретический материал

**Автоматизированная система управления** или **АСУ** – комплекс аппаратных и программных средств, предназначенный для управления различными процессами в рамках технологического процесса, производства, предприятия. АСУ применяются в различных отраслях промышленности, энергетике, транспорте и тому подобное.

**Важнейшая задача АСУ**– повышение эффективности управления объектом на основе роста производительности труда и совершенствования методов планирования процесса управления.

#### Цели автоматизации управления

Обобщенной целью автоматизации управления является повышение эффективности использования потенциальных возможностей объекта управления. Таким образом, можно выделить ряд целей:

1. Предоставление лицу, принимающему решение (ЛПР) адекватных данных для принятия решений.
2. Ускорение выполнения отдельных операций по сбору и обработке данных.
3. Снижение количества решений, которые должно принимать ЛПР.
4. Повышение уровня контроля и исполнительской дисциплины.
5. Повышение оперативности управления.
6. Снижение затрат ЛПР на выполнение вспомогательных процессов.
7. Повышение степени обоснованности принимаемых решений.

В состав АСУ входят следующие **виды обеспечений**:

- |                    |                    |
|--------------------|--------------------|
| – информационное,  | – метрологическое, |
| – программное,     | – правовое,        |
| – техническое,     | – лингвистическое. |
| – организационное, |                    |

#### Основные классификационные признаки

Основными классификационными признаками, определяющими вид АСУ, являются:

- сфера функционирования объекта управления (промышленность, строительство, транспорт, сельское хозяйство, непромышленная сфера и так далее);
- вид управляемого процесса (технологический, организационный, экономический и так далее);
- уровень в системе государственного управления, включения управление народным хозяйством в соответствии с действующими схемами управления отраслями (для промышленности: отрасль (министерство), всесоюзное объединение, всесоюзное промышленное объединение, научно-производственное объединение, предприятие (организация), производство, цех, участок, технологический агрегат).

#### Функции АСУ

Функции АСУ в общем случае включают в себя следующие элементы (действия):

- планирование и (или) прогнозирование;

- учет, контроль, анализ;
- координацию и (или) регулирование.

**Виды АСУ**

- Автоматизированная система управления технологическим процессом или АСУ ТП– решает задачи оперативного управления и контроля техническими объектами в промышленности, энергетике, на транспорте.
- Автоматизированная система управления производством (АСУ П)– решает задачи организации производства, включая основные производственные процессы, входящую и исходящую логистику. Осуществляет краткосрочное планирование выпуска с учётом производственных мощностей, анализ качества продукции, моделирование производственного процесса.

**Примеры:**

- Автоматизированная система управления уличным освещением («АСУ УО»)– предназначена для организации автоматизации централизованного управления уличным освещением.
- Автоматизированная система управления наружного освещения («АСУ-НО»)– предназначена для организации автоматизации централизованного управления наружным освещением.
- Автоматизированная система управления дорожным движением или АСУ ДД– предназначена для управления транспортных средств и пешеходных потоков на дорожной сети города или автомагистрали
- Автоматизированная система управления предприятием или АСУП– Для решения этих задач применяются MRP,MRP II и ERP-системы.
- Автоматическая система управления для гостиниц.
- Автоматизированная система управления операционным риском– это программное обеспечение, содержащее комплекс средств, необходимых для решения задач управления операционными рисками предприятий: от сбора данных до предоставления отчетности и построения прогнозов.

Схема «Виды АСУ»

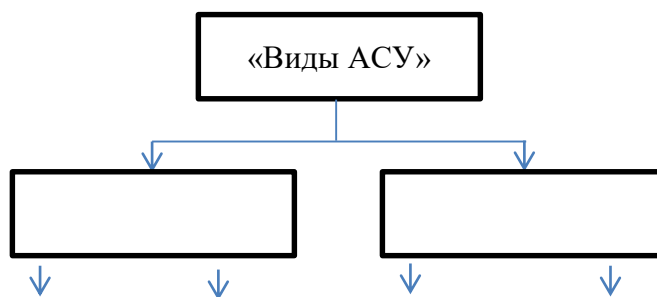


Таблица 5

Определение АСУ	
Какие виды обеспечений входят в состав АСУ	
Виды АСУ	
Функции АСУ	
Примеры автоматизированных систем	

управления их краткая характеристика	
Область применения АСУ	

## Практическая работа №33 Работа в браузере MS Internet Explorer

**Цель:** технологию работы с браузером MS Internet Explorer

### Теоретический материал

**Браузер** – программа, служащая для просмотра Web – документов, т. е. обеспечивающая переход на другой объект в соответствии с гиперссылкой.

**Гиперссылка** – выделенный объект, связанный с другим файлом и реагирующий на щелчок мыши.

**Гипертекст** – документ, содержащий ссылки на другие документы.

Техника чтения обычного текста состоит в том, что, закончив чтение одной страницы, вы приступаете к следующей. Технология же гипертекста позволяет вам свободно переходить со страницы на страницу, следуя заинтересовавшим вас ссылкам, - никакого заранее установленного порядка чтения не существует.

Чаще ссылка выделяется на фоне экрана монитора с помощью изменения цвета и подчеркивания. В таком случае о ней говорят "гиперссылка" или даже "гиперсвязь".

Гиперссылкой можно считать слово, фразу или графический элемент, если указатель мыши, установленный на этом объекте, меняет свое изображение.

Гипертекст с помощью гиперссылок связывает между собой множество документов. WWW связывает миллионы гипертекстовых документов, которые в данном случае называются – Web-документами, или Web-страницами, т. к. содержат не только текстовые данные, но и графические, анимационные, музыкальные.

Web-документы хранятся в файлах. Создаются эти файлы с помощью одних программ, а просматриваются с помощью программ просмотра (браузеров). Существует много программ-браузеров. У них одна основная функция – обеспечить отображение Web-документов.

Один из известных браузеров отображения Web-страниц – Internet Explorer.

### ХОД РАБОТЫ

1. Запустите приложение Internet Explorer, обычно ярлык программы находится на «Рабочем столе»
2. Рассмотрите окно браузера. Поле заголовка окна содержит название загруженной страницы, поэтому оно должно быть написано на английском языке или латинскими буквами. Строка состояния отображает ход загрузки данных или состояние работы браузера. Узнать эту строку легко – она расположена в нижней части главного окна.
3. Ознакомьтесь с функцией строки Адрес – в ней вы сможете прочитать полный адрес загруженного браузером документа. Строка Адрес находится над областью просмотра главного окна. Пользователи Интернета чаще вместо термина «адрес» употребляют аббревиатуру URL – Uniform Resource Location – местонахождение Web-документа в сети, хотя разумнее термин «адрес» соотносить с компьютером, а о находящемся на нем документе говорить «местонахождение». URL – это комбинация названия протокола передачи документа, адреса сервера, каталога и имени файла. URL – адрес документа в сети.
4. Ознакомьтесь с меню Избранное. Оно позволяет создавать закладки, управлять ими. Закладка создается указанием браузеру запомнить необходимый, весьма



длинный URL (адрес). Для этого надо щелкнуть левой кнопкой мыши по соответствующему пункту основного меню и в открывшемся подменю выбрать Добавить в «Избранное»...

5. Ознакомьтесь с некоторыми кнопками главного окна браузера Internet Explorer. Панель инструментов находится ниже строки основного меню и включает часто используемые команды. Её функция – дать пользователю возможность быстро выбрать команду меню с помощью мыши. Кнопки Панели инструментов снабжены текстовыми надписями. Как принято в Windows, недоступные кнопки отображаются на экране серым цветом. Кнопка Избранное . Если щелкните по этой кнопке, в левой части основного окна просмотра документа откроется новое окно со списком всех имеющихся закладок, одну из которых можно выбрать щелчком левой кнопки мыши.
6. Посетите все страницы в обратном порядке, нажимая для этого на кнопку меню Назад
7. Сделайте анализ достоинств и недостатков такого обслуживания Web.

### Практическая работа №34 HTML. Инструментарий. Основные понятия

**Цель:** изучить инструментарий и основные понятия языка HTML/

#### Теоретический материал

Прежде чем создавать сайт, надо понять из чего собственно он состоит. Когда вы набираете в адресной строке браузера, например [www.yandex.ru](http://www.yandex.ru), вы знаете, что попадете на сайт с именем yandex. Далее, щелкая по ссылкам и кнопочкам, вы переходите со страницы на страницу этого ресурса.

Отсюда нетрудно догадаться, что сайт - это набор страниц, связанных между собой. Сделать сайт - значит научиться делать и связывать эти странички.

#### ХОД РАБОТЫ

1. Откройте блокнот (Notepad) и введите в новый документ следующий код:

```
<html>
<head>
  <title> Моя первая страница </title>
</head>
<body>
  Всем привет!
</body>
</html>
```

2. Сохраните этот документ (Файл -> Сохранить как), укажите место хранения вашего файла (у меня - это диск D). Имя файла - index.html, тип файла - все файлы.
3. Запустите браузер (Пуск -> Internet Explorer), в адресной строке укажите полный путь к созданному файлу index.html.

Посмотрите на код, который мы вводили в блокноте. Это и есть структура любого html-документа. Все, что находится между **<** и **>** называется тег.

Между **<** и **>** находится имя тега и его параметры (если они есть).

В основном все теги парные, т.е. один открывающий **<>**, а другой - закрывающий **</>**. Такая пара тегов называется контейнером. А все, что находится между ними - содержимым контейнера. Действия тегов распространяются только на их содержимое.

Бывают также теги одиночные, т.е. не требующие закрытия. Такие теги просто совершают определенное действие, например, перевод строки или вставку рисунка.

- в теги <html> </html> помещается весь документ
- в теги <head> </head> помещается разная информация для браузера (об этом позже)
- в теги <title> </title> помещается заголовок страницы (он виден на самом верху страницы)
- в теги <body> </body> помещается все, что вы хотите разместить (картинки и текст)

#### 4. Измените фон html-страницы и цвет текста.

Для этого добавим в наш код, в тег <body> следующие атрибуты - `bgcolor="blue" text="yellow"`. Здесь `bgcolor` - атрибут, отвечающий за цвет фона страницы, а `blue` - его значение (в нашем случае - голубой, но можете сделать и красный - `red`, и зеленый - `green`, и любой другой). Атрибут `text` задает цвет текста документа, его значение `yellow` - желтый.

```
<html>
  <head>
    <title> Моя первая страница </title>
  </head>
  <body bgcolor="blue" text="yellow">
    Всем привет!
  </body>
</html>
```

#### 5. Запустите браузер (как в шаге 3 первого урока), теперь ваша страница выглядит так:

Обратите внимание, что атрибуты, указанные в теге <body> распространяют свое действие на весь документ, т.е. весь введенный вами текст будет теперь сиреневого цвета, что не очень удобно. Поэтому лучше убрать атрибут `text` из тега <body>, тогда цвет текста будет установлен по умолчанию, т.е. черный. А задавать цвет различных частей текста лучше в самом тексте.

Для этого добавим несколько предложений, заключим их в нужные теги, а затем разберемся, что они делают.

```
<html>
  <head>
    <title> Моя первая страница </title>
  </head>
  <body bgcolor="#CCCCFF">
    <h1>
      <font color="red">
        Это заголовок первого уровня
      </font>
    </h1>
    Это простой текст.
    <b>Этот выделен полужирным начертанием.</b>
      <br>
      <i>А этот курсивом.</i>
    <h2>
      <font color="red">
```

```

    Это заголовок второго уровня
    </font>
  </h2>
  <font color="#996666" size="5" face="Arial">
    Это текст Arial, размер шрифта - 5
  </font>
  <h3>
    <font color="red">
      Это заголовок третьего уровня
    </font>
  </h3>
  <font color="green" size="2" face="Verdana">
    Этот текст Verdana, размер шрифта - 2
  </font>
</body>
</html>

```

6. Обновите html-страницу (Ctrl+F5) и посмотрите на результат.

- 1) Теги `<h1> </h1>`, `<h2> </h2>` и т.д. называются заголовками. HTML поддерживает 6 уровней заголовков от h1 до h6. h1 - самый главный и крупный, далее по убыванию значения и шрифта до h6 - самого маленького и по значению, и по размеру шрифта.
- 2) Теги `<font> </font>` управляют шрифтом. Для этого используются атрибуты:
- 3) - *color* - задает цвет текста,
- 4) - *size* - задает размер шрифта (по умолчанию равен 3),
- 5) - *face* - задает гарнитуру шрифта. В принципе, вы можете указывать здесь любой понравившийся вам шрифт, но учтите, что на компьютере пользователя такого шрифта может и не оказаться. Поэтому рекомендуется использовать стандартные: Times New Roman, Arial, Verdana и Tahoma.

Теги `<b> </b>` делают текст жирным.

- 1) Теги `<i> </i>` отображают текст курсивом.
- 2) Тег `<br>` начинает текст с новой строки. Обратите внимание, этот тег одиночный, т.е. не имеет закрывающего тега.

### Практическая работа №35 HTML. Таблицы.

**Цель:** освоение приемов создания таблиц на языке HTML

#### Теоретический материал

Таблица - один из основных инструментов для создания web-страниц. Без использования CSS, только с помощью таблиц можно создавать страницы со сложным дизайном. Если вы прошли серию уроков [Делаем сайт - первые шаги](#), то вы понимаете о чем речь. Любая таблица представляет собой набор строк и столбцов, на пересечении которых находятся ячейки.

1. Создать таблицу

Название таблицы		
столбец 1	столбец 2	столбец 3

первый столбец первой строки	второй столбец первой строки	третий столбец первой строки
первый столбец второй строки	второй столбец второй строки	третий столбец второй строки
первый столбец третьей строки	второй столбец третьей строки	третий столбец третьей строки

- сама таблица задается с помощью тегов <table></table>,
- у таблицы есть название - теги <caption></caption>,
- таблица состоит из строк - теги <tr></tr>,
- каждая строка состоит из столбцов - теги <td></td>,
- столбцы имеют названия, расположенные в первой строке - теги <th></th>.

```

<html>

<head>
  <title>html table</title>
</head>

<body>

  <table>
    <caption>Название таблицы</caption>
    <tr><th>1</th><th>2</th><th>3</th></tr>
    <tr><td>11</td><td>12</td><td>13</td></tr>
    <tr><td>21</td><td>22</td><td>23</td></tr>
    <tr><td>31</td><td>32</td><td>33</td></tr>
  </table>

</body>

</html>

```

2. Оформление таблицы: Для этого у тега <table> существует ряд параметров:
  - width - задает ширину таблицы (в пикселах или % от ширины экрана),
  - 
  - bgcolor - задает цвет фона ячеек таблицы,
  - 
  - background - заливает фон таблицы рисунком,
  - 
  - border - задает рамку указанной ширины (в пикселах) вокруг всей таблицы.
3. Примените эти параметры:

```

<html>

<head>
  <title>html table</title>
</head>

<body>

  <table width="300" bgcolor="plum" border="1">
    <caption>Название таблицы</caption>

```

```
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td>11</td><td>12</td><td>13</td></tr>
<tr><td>21</td><td>22</td><td>23</td></tr>
<tr><td>31</td><td>32</td><td>33</td></tr>
</table>
```

```
</body>
```

```
</html>
```

4. Измените параметры:

- align - задает выравнивание таблицы: слева (right), справа (left), по центру (center),
- 
- cellspacing - задает расстояние между ячейками таблицы (в пикселах),
- 
- cellpadding - задает расстояние между текстом и внутренней границей ячейки таблицы (в пикселах).

```
<html>
```

```
<head>
```

```
<title>html table</title>
```

```
</head>
```

```
<body>
```

```
<table width="300" bgcolor="plum" border="1"
align="center" cellspacing="5" cellpadding="10">
```

```
<caption>Название таблицы</caption>
```

```
<tr><th>1</th><th>2</th><th>3</th></tr>
```

```
<tr><td>11</td><td>12</td><td>13</td></tr>
```

```
<tr><td>21</td><td>22</td><td>23</td></tr>
```

```
<tr><td>31</td><td>32</td><td>33</td></tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

5. Сохраните результат.

### Практическая работа №36 HTML.Фреймы

**Цель:** освоение приемов создания web-страниц на языке HTML с применением фреймов

#### ХОД РАБОТЫ

Формат документа, использующего фреймы, внешне очень напоминает формат обычного документа, только вместо тэга **BODY** используется контейнер **FRAMESET**, содержащий описание внутренних HTML-документов, содержащий собственно информацию, размещаемую во фреймах.

1. Откройте блокнот, наберите код:  

```
<HTML>
<HEAD>...</HEAD>
<FRAMESET>...</FRAMESET>
</HTML>
```
2. Сохраните файл в специальном формате (укажите тип файла «Все файлы», расширение .html)/
3. Отредактируйте код:  

```
<FRAMESET COLS="value" | ROWS="value">
<FRAME SRC="url1">
<FRAME ...>
...
</FRAMESET>
```

Общий контейнер FRAMESET описывает все фреймы, на которые делится экран. Вы можете разделить экран на несколько вертикальных или несколько горизонтальных фреймов. Тэг FRAME описывает каждый фрейм в отдельности. Рассмотрим более детально каждый компонент.

Тег FRAMESET.

```
<FRAMESET [COLS="value" | ROWS="value"]>
```

Тэг <FRAMESET> имеет завершающий тэг </FRAMESET>. Все, что может находиться между этими двумя тэгами, это тэг <FRAME>, вложенные тэги <FRAMESET> и </FRAMESET>, а также контейнер из тэгов <NOFRAME> и </NOFRAME>, который позволяет строить двойные документы для браузеров, поддерживающих фреймы и не поддерживающих фреймы.

Данный тэг имеет два взаимоисключающих параметра: ROWS и COLS.

ROWS="список-определений-горизонтальных-подокон"

Данный тэг содержит описания некоторого количества подокон, разделенные запятыми. Каждое описание представляет собой числовое значение размера подокна в пикселах, процентах от всего размера окна или связанное масштабное значение. Количество подокон определяется количеством значений в списке. Общая сумма высот подокон должна составлять высоту всего окна (в любых измеряемых величинах). Отсутствие атрибута ROWS определяет один фрейм, величиной во все окно браузера.

Синтаксис используемых видов описания величин подокон:

value

Простое числовое значение определяет фиксированную высоту подокна в пикселах. Это далеко не самый лучший способ описания высоты подокна, поскольку различные браузеры имеют различный размер рабочего поля, не говоря уже о различных экранных разрешениях у пользователя. Если вы, все же, используете данный способ описания размера, то настоятельно рекомендуется сочетать его с каким-либо другим, чтобы в результате вы точно получили 100%-ное заполнение окна браузера вашего пользователя.

value%

Значение величины подокна в процентах от 1 до 100. Если общая сумма процентов описываемых подокон превышает 100, то размеры всех фреймов пропорционально уменьшаются до суммы 100%. Если, соответственно, сумма меньше 100, то размеры пропорционально увеличиваются.

value\*

Вообще говоря, значение value в данном описании является необязательным. Символ "\*" указывает на то, что все оставшееся место будет принадлежать данному фрейму. Если указывается два или более фрейм с описанием "\*" (например "\*", "\*"), то оставшееся пространство делится поровну между этими фреймами. Если перед звездочкой стоит цифра, то она указывает пропорцию для данного фрейм (во сколько раз он будет больше аналогично описанного чистой звездочкой). Например, описание "3\*,\*,\*" говорит, что будет

создано три фрейм с размерами 3/5 свободного пространства для первого фрейм и по 1/5 для двух других.

4. Создавайте три файла

Файл `index.html`, Файл `fr_1.html`, Файл `fr_2.html`

5. В файле: `index.html` введите следующий код:

```
<HTML>
<HEAD> </HEAD>
  <FRAMESET COLS="30%,*">
    - разбиваем окно на 2-а подокна по вертикали (т.е. описываем два фрейм), 1-ое подокно занимает 30% от экрана, 2-ое подокно занимает оставшуюся часть экрана.
    <FRAME SRC="fr_1.html" NAME="left" noresize >
      - в 1-ом подокне (в левом) отображаем файл fr_1.html, имя фрейма left, размер фрейма изменять нельзя
    <FRAME SRC="fr_2.html" NAME="right" >
      - в 2-ом подокне (в правом) отображаем файл fr_2.html, имя фрейма right
  </FRAMESET>
</HTML>
```

6. В файле: `fr_1.htm` введите следующий код:

```
<HTML>
<HEAD> </HEAD>
<BODY>
  <H1 ALIGN="CENTER">МАШИНА</H1>
  <DIV ALIGN="CENTER"><IMG SRC="11_Auto.jpg" WIDTH=200
HEIGHT=200></DIV>
</BODY>
</HTML>
```

7. В файле: Файл `fr_2.html` введите следующий код:

```
<HTML>
<HEAD> </HEAD>
<BODY>
  <H1 ALIGN="CENTER">КОТЯТА</H1>
  <DIV ALIGN="CENTER"><IMG SRC="11_ket.JPG" WIDTH=300
HEIGHT=300></DIV>
</BODY>
</HTML>
```

8. Сохраните результат проверьте работу фреймов.

### Практическая работа №37 Осуществление поиска информации в сети Интернет

**Цель:** Изучение поисковых систем технологии поиска информации в сети Интернет.

#### Теоретический материал

**Поисковая система** – это комплекс программ и мощных компьютеров, способные принимать, анализировать и обслуживать запросы пользователей по поиску информации в Интернет. Поскольку современное Web-пространство необозримо, поисковые системы вынуждены создавать свои базы данных по Web- страницам. Важной задачей поисковых систем является постоянное поддержание соответствия между созданной информацией-

ной базой и реально существующими в Сети материалами. Для этого специальные программы (роботы) периодически обходят имеющиеся ссылки и анализируют их состояние. Данная процедура позволяет удалять исчезнувшие материалы и по добавленным на просматриваемые страницы ссылкам обнаруживать новые.

При наличии первичных сведений по теме поиска, документы можно искать в поисковых системах. При этом следует различать приемы простого, расширенного, контекстного и специального поиска.

Под **простым поиском** понимается поиск Web-ресурсов по одному или нескольким ключевым словам. Недостаток простого поиска заключается в том, что обычно он выдает слишком много документов, среди которых трудно выбрать наиболее подходящие.

При использовании **расширенного поиска** ключевые слова связывают между собой операторами логических отношений. Расширенный поиск применяют в тех случаях, когда приемы простого поиска дают слишком много результатов. С помощью логических отношений поисковое задание формируют так, чтобы более точно детализировать задание и ограничить область отбора, например по дате публикации или типу данных.

**Контекстный поиск** – это поиск по точной фразе. Он удобен для реферативного поиска информации, но доступен далеко не во всех поисковых системах. Прежде всего, чтобы обеспечивать такую возможность, система должна работать не только индексированными файлами, но и с полноценными образами Web-страниц. Эта операция достаточно медленная, и ее выполняют не все поисковые системы.

**Специальный поиск** применяют при поиске Web-страниц, содержащих ссылки на заданные адреса URL, содержащих заданные данные в служебных полях, например в поле заголовка и т.п.

#### ХОД РАБОТЫ

1. Загрузите Интернет.
2. С помощью строки поиска найдите каталог поисковых систем.
3. Выпишите электронные адреса поисковых систем (не менее 5) и дайте им краткую характеристику. Оформите в виде таблицы 12:

Таблица 12

№	Название поисковой системы	Электронный адрес поисковой системы	Характеристика поисковой системы

4. Загрузите страницу электронного словаря [www.gumer.info/bibliotek\\_Buks/Econom](http://www.gumer.info/bibliotek_Buks/Econom) и заполните таблицу 13.

Таблица 13

Термин	Определение
Актив	
Аффилированные лица	
Единица бухгалтерского учета	
Метод начисления	
Не денежные средства	
Непокрытый убыток	
Нематериальные активы	
Резервный капитал	
Уставный фонд	
Резервы	



Регистры бухгалтерского учета	
Нетто-оценка	
Учетная политика организации	

**Расширенный поиск.** Кроме средства простого поиска обычно поисковые службы предоставляют средства расширенного поиска. Эти средства позволяют более точно формулировать поисковое задание, но требуют определенного опыта и работают заметно медленнее. В большинстве поисковых систем команды расширенного поиска формируются с помощью логических команд. Удобство использования логических команд в частности связано с тем, что команды простого поиска у многих поисковых систем реализованы по-разному. Каждая система стремится сделать средства простого поиска наиболее удобными, а средства расширенного поиска – наиболее стандартными. Тем не менее, для обозначения логических операторов в различных поисковых системах используются разные обозначения.

5. Используя таблицу 12 пройдите по адресам поисковых систем и изучите синтаксис поисковых запросов. Результаты оформите в таблицу 14.

Таблица 14

Поисковая система	Оператор	Описание

### Практическая работа №38 Создание ящика электронной почты и настройка его параметров

**Цель:** изучение технологии создания ящика электронной почты, настройка его параметров.

#### Теоретические материалы

*Электронная почта* – одна из наиболее распространенных и популярных функций компьютерных сетей, обеспечивающая обмен сообщениями между пользователями сети.

По электронной почте можно пересылать не только текстовые сообщения, но и готовые файлы, созданные в любых других программах.

Существует большое количество WWW-серверов, которые предлагают завести бесплатный почтовый ящик и позволяют работать с почтой, используя только браузер. Чтобы получить бесплатный почтовый ящик на таком сервере, необходимо зарегистрироваться.

## ХОД РАБОТЫ

1. Регистрация на бесплатном почтовом сервере.
  - Запустите интернет-браузер, зайдите на сайт [www.yandex.ru](http://www.yandex.ru).
  - Выберите ссылку **«Почта – Зарегистрироваться»**, заполните форму регистрации. При введении имени и фамилии будут предложены автоматически свободные логины, можно выбрать из предложенных или придумать собственный, который будет проверен почтовым сервером, занят ли он другим пользователем.
  - поля **«Логин»**, **«Пароль»** и **«Подтверждение»** пароля должны заполняться латинскими буквами, причем пароль должен содержать не менее 4-х символов;
  - обязательные поля для заполнения отмечены звездочками.
  - Подтвердите данные, нажав кнопку **«Зарегистрироваться»**
2. Откройте свой новый почтовый ящик на бесплатном почтовом сервере: для этого введите свой логин и пароль, нажмите кнопку **«Вход»**. После этого откроется ваша почта, которая будет содержать основные элементы:
  - **«Входящие»** содержит всю поступившую к вам почту.
  - **«Отправленные»** содержит всю отправленную вами другим адресатам почтовые сообщения.
  - **«Рассылки»** складываются письма, которые были одновременно разосланы большому числу пользователей.
  - **«Удаленные»** содержат удаленные письма из любой другой папки.
  - **«Черновики»** содержат не отправленные письма.
3. Работа с почтовыми сообщениями.
  - Создайте сообщение с темой **«Отчетная документация»**, щелкните по кнопке **«Написать»**. Заполните заголовки сообщения: в заголовке **«Кому»** укажите адрес преподавателя [habira32@mail.ru](mailto:habira32@mail.ru), в тексте сообщения впишите свои фамилию, имя, отчество, номер группы. Отправьте сообщение с помощью кнопки **«Отправить»**.
  - Перейдите в папку **«Входящие»**. Вам должно прийти сообщение преподавателя. Для того, чтобы прочитать полученное сообщение, необходимо нажать на ссылку в поле **«От кого»**.
  - В появившемся окне нажмите на кнопку **«Ответить»**. Напишите ответ на это письмо и нажмите на кнопку **«Отправить»**.
  - Создайте новое сообщение и вложите в него текстовый файл:
  - На рабочем столе правой кнопкой мыши создайте документ Microsoft Word, назовите «Список документов для открытия ИП», наберите текст который будет содержать список документов для открытия ИП (можно воспользоваться интернетом), закройте файл, сохраните. Вернитесь в свой электронный ящик, щелкните по кнопке **«Написать»**. Заполните заголовки сообщения: Кому, Тема следующим образом: в заголовке Кому укажите адрес преподавателя. В качестве Темы укажите «Список документов».
  - Нажмите на кнопку **«Обзор»**, укажите местонахождение файла (напишите текст сообщения). Отправьте сообщение, нажав на соответствующую кнопку.
  - Создайте новое сообщение и вложите в него графический файл (аналогичным образом так же как и текстовый):
4. Заполнение адресной книги.
  - Занесите в Адресную книгу новых абонентов. Пополните **«Адресную книгу»**, воспользовавшись пунктом меню **«Сервис - Адресная книга»** или соответствующей кнопкой на панели инструментов.

- Внесите в Адресную книгу преподавателя, соседа справа и слева. Для этого выполните команду «**Файл - Создать контакт**» (или щелкните левой кнопкой мыши на кнопке «**Создать**» и выберите пункт меню «**Создать контакт**»). Внимательно изучите вкладки, представленные в данном диалоговом окне. Обратите внимание на то, что в нем имеются средства для ввода как личной, так и служебной информации (для практической деятельности, как правило, достаточно заполнить лишь несколько полей на вкладке **Имя**).
- Начните заполнение полей вкладки «**Имя**» с поля «**Имя в книге**». Введите сюда такую запись, которую хотели бы видеть в списке контактов. Заполните поля **Фамилия**, **Имя** и **Отчество**.
- В поле «**Адреса электронной почты**» введите его электронный адрес.
- Занесите введенные данные в «**Адресную книгу**», нажав на кнопку «**Добавить**». Если необходимо изменить внесенные данные, следует щелкнуть на записи правой кнопкой мыши, в контекстном меню выбрать пункт «**Свойства**» и перейти на вкладку «**Имя**».
- Самостоятельно отправьте несколько сообщений с вложенными текстовыми и графическими файлами. Внесите в адресную книгу несколько новых контактов.

### **Практическая работа №39 Использование тестирующих систем**

**Цель:** Ознакомиться со средой тестирующей системы MyTestX

#### **ХОД РАБОТЫ**

1. Пройти тест в программе MyTestX:
  - Откройте папку MyTest → MyTestStudent (зеленый квадратик). В открывшейся программе есть 2 способа запустить тест:  
Файл → Открыть (далее выбрать файл с тестом);  
Файл → Получить по сети (далее выбрать тест из списка).
  - Выберите второй способ откроется тест «Демонстрационный тест №1». прочитайте «Информацию», после чего «Для начала прохождения теста дайте команду Тест → Начать тест...».В открывшемся окне «Начнем тест?!» вводите свою Фамилию, Имя и курс. Нажмите ОК.
  - При прохождении теста обратите внимание на 10 различных типов вопросов. Пройдите тест до конца и покажите результат преподавателю.
2. Создать тест в программе MyTestX:
  - В папке MyTest → Tests → Информатика выбираем вариант.
  - Откройте MyTest → MyTestEditor (голубой квадратик).
  - Пустой тест создан, зайдите в меню: Параметры теста → Заголовок и описание...
  - В открывшемся окне ввести заголовок (у всех один); ввести свои Фамилию Имя; в описание теста ввести свою тему. Нажать кнопку «Сохранить изменения».
  - Добавить первый вопрос: Задание → Добавить... Одиночный выбор → ОК

### Заполняем «Задание #1»

рис 7:

- В окне «Основное» скопируйте вопрос (без номера);
- Скопируйте варианты ответа (без номера, «;», «.»);
- В конце теста есть ключ, выберите верный ответ;
- Нажмите «Сохранить задание».

3. Добавьте **12-15 заданий** и сохраните тест в папке **MyTest** → **Tests**.

4. Открыть первым способом созданный тест и пройти его.

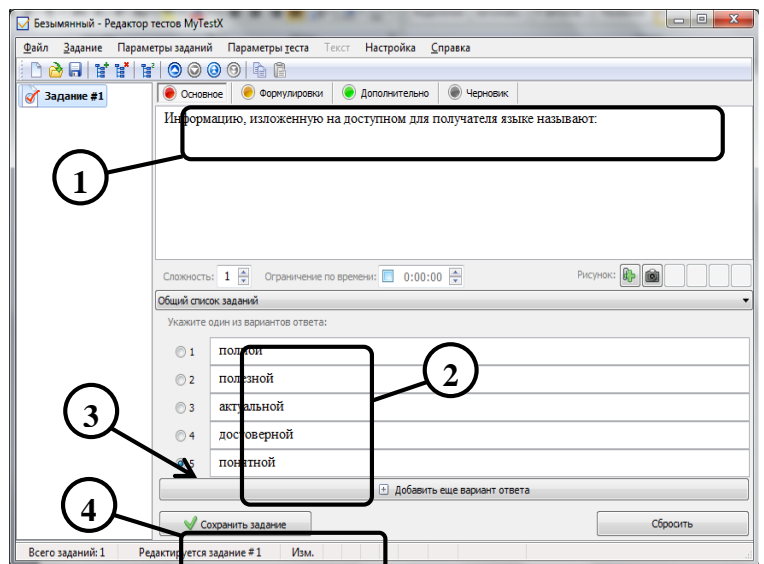


Рис. 7

## ЛИТЕРАТУРА

### Основные источники:

1. Кравченко, Л.В. Практикум по Microsoft office 2007 (Word, Excel, Access), PhotoShop [Электронный ресурс]: учебно-методическое пособие/ Л.В. Кравченко – 2 изд., испр. и доп.- Москва: Форум, ИНФРА, 2015. -168с., <http://znanium.com/bookread2.php?book=478844>
2. Синаторов, С.В. Информационные технологии [Электронный ресурс]: учебное пособие / С.В. Синаторов.- Москва: ФЛИНТА, 2016.- 448с. <https://e.lanbook.com/reader/book/83798/#2>

### Дополнительные источники:

1. Плотникова, Н.Г. Информатика и информационно-коммуникационные технологии (ИКТ) [Электронный ресурс]: учеб. пособие. — М.: РИОР: ИНФРА-М, 2017. — 124 с., <http://znanium.com/bookread2.php?book=760298>

### Интернет-ресурсы:

1. [alleng.ru/edu/comp.htm](http://alleng.ru/edu/comp.htm) (Образовательные ресурсы Интернета - Информатика)
2. [school-collection.edu.ru](http://school-collection.edu.ru) (Единая коллекция цифровых образовательных ресурсов).
3. [www.intuit.ru/studies/courses](http://www.intuit.ru/studies/courses) (Открытые интернет-курсы «Интуит» по курсу «Информатика»)
4. [www.gosuslugi.ru](http://www.gosuslugi.ru) (Портал государственных услуг)
5. [www.klyaksa.net](http://www.klyaksa.net) (Информационно-образовательный портал)