

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИ ВлГУ)**

ИНФОРМАТИКА
**Методические указания по выполнению лабораторных работ
для студентов образовательной программы
11.02.01 Радиоаппаратостроение.**

**Составитель:
Якименко К.А.**

**г. Муром
2018**

Данная работа содержит методические указания к лабораторным работам по дисциплине «Информатика» и предназначена для обучающихся по специальностям среднего профессионального образования.

Цель разработки: оказание помощи обучающимся в выполнении лабораторных работ по предмету «Информатика».

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ | 5 |
| Лабораторная работа № 1. ОСНОВЫ РАБОТЫ В WINDOWS. ФАЙЛОВАЯ СТРУКТУРА. | 6 |
| Лабораторная работа № 2. ОСНОВЫ РАБОТЫ В MS WORD. СОЗДАНИЕ ТЕКСТОВЫХ ДОКУМЕНТОВ. | 11 |
| Лабораторная работа № 3. ОСНОВЫ РАБОТЫ В MS WORD. ФОРМАТИРОВАНИЕ ТЕКСТОВЫХ ДОКУМЕНТОВ. | 14 |
| Лабораторная работа № 4. ОСНОВЫ РАБОТЫ В MS EXCEL. СОЗДАНИЕ И ФОРМАТИРОВАНИЕ ЭЛЕКТРОННЫХ ТАБЛИЦ..... | 16 |
| Лабораторная работа № 5. ОСНОВЫ РАБОТЫ В MS EXCEL. ВЫЧИСЛЕНИЯ В MS EXCEL. | 19 |
| Лабораторная работа № 6. ОСНОВЫ РАБОТЫ В MS EXCEL. СОЗДАНИЕ ГРАФИКОВ В MS EXCEL. | 22 |
| Лабораторная работа № 7. ОСНОВЫ РАБОТЫ В MS ACCESS. СОЗДАНИЕ БАЗЫ ДАННЫХ..... | 25 |
| Лабораторная работа № 8. СОЗДАНИЕ ПРЕЗЕНТАЦИЙ В MS POWER POINT..... | 29 |
| Лабораторная работа № 9. ИЗУЧЕНИЕ ГРАФИЧЕСКИХ РЕДАКТОРОВ. | 32 |
| Лабораторная работа № 10. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ..... | 35 |
| Лабораторная работа № 11. ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ..... | 40 |
| Лабораторная работа № 12. ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ. | 43 |
| Лабораторная работа № 13. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ОДНОМЕРНЫХ МАССИВОВ. | 45 |
| Лабораторная работа № 14. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ДВУМЕРНЫХ МАССИВОВ. | 48 |
| Лабораторная работа № 15. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СТРУКТУР..... | 50 |
| Лабораторная работа № 16. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СТРОК. | 53 |
| Лабораторная работа № 17. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ..... | 56 |

| | |
|--|----|
| Лабораторная работа № 18. ПОИСК ИНФОРМАЦИИ В ГЛОБАЛЬНОЙ СЕТИ: КАТАЛОГАХ, ЭЛЕКТРОННЫХ БИБЛИОТЕКАХ И СПРАВОЧНИКАХ..... | 58 |
| Лабораторная работа № 19. ОСНОВЫ ЯЗЫКА HTML..... | 60 |
| Лабораторная работа № 20. СОЗДАНИЕ WEB-САЙТА НА ЯЗЫКЕ HTML..... | 64 |
| СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ..... | 66 |

ВВЕДЕНИЕ

Методические указания предназначены для обучающихся, изучающих основы работы на персональных компьютерах в рамках курса «Информатика». Они используются при изучении разделов данного курса: «Компьютер и программное обеспечение», «Коммуникационные технологии».

В методические указания включены двадцать лабораторных работ, предназначенных для приобретения обучающимися первоначальных теоретико-практических знаний и навыков решать свои производственные работы на персональном компьютере, а в период обучения – выполнять на компьютере информационно-вычислительные работы по другим учебным дисциплинам.

Лабораторная работа № 1. ОСНОВЫ РАБОТЫ В WINDOWS. ФАЙЛОВАЯ СТРУКТУРА.

Цель работы: изучить основные принципы работы в операционной системе Windows: приёмы управления при помощи мыши, операций с файлами при помощи системы окон папок и работы организацию файловой структуры при помощи программы Проводник.

1. Приемы управления при помощи мыши

Управлять ОС Windows можно с помощью клавиатуры и устройства мышь.

Мышь – координатное устройство для управления курсором и отдачи различных команд компьютеру. Как правило, мышь имеет две кнопки для управления (левую и правую) и колесо, расположенное посередине. Колесо служит для управления полосами прокрутки. Мышь позволяет управлять виртуальным элементом на экране компьютера – курсором, представленным в виде стрелки.

Основными приемами управления с помощью мыши являются:

1) **наведение указателя мыши** на экранный объект без использования кнопок. Этот прием позволяет вызвать всплывающую подсказку, открывать вложенные меню.

2) **щелчок** – это быстрое нажатие и отпускание основной (по умолчанию – левой) кнопки мыши. Этот прием позволяет:

- выделить экранный объект (значок, текст);
- нажать экранную кнопку;
- выбрать команду в меню.

3) **двойной щелчок** – это два щелчка, быстро следующих друг за другом без какой-либо паузы. Этим приемом можно открыть документ или запустить программу.

4) **щелчок правой (дополнительной) кнопкой** мыши открывает контекстное меню, относящееся к экранному объекту.

Контекстное меню – это список действий, которые можно применить к объекту в конкретной ситуации. Можно выбрать конкретный пункт меню, щёлкнув его мышью. Если щёлкнуть мышью в любом месте вне контекстного меню, то оно будет убрано с экрана.

5) **перетаскивание** – наведя указатель на экранный объект, нажмите основную кнопку мыши и, не отпуская ее, двигайте мышь. Объект начнет двигаться вместе с указателем. Перетаскивание используют для перемещения или копирования объектов, для изменения размеров или формы объектов, а также в операциях выделения.

6) **поворот колесика** помогает при просмотре больших документов. Если содержимое не полностью помещается на экран, поворот колесика позволяет прокручивать документ. Чаще всего эту возможность используют при работе с документами Интернета.

7) **щелчок колесиком.** Стандартной функции за этим приемом не закреплено, но в некоторых программах эта возможность используется. Например, есть функция плавного регулирования скорости просмотра изображений и текстов.

2. Операции с файловой системой WINDOWS при помощи системы окон папок

Файл – основной носитель информации в операционной системе, представляющий собой физическую область на диске, имеющий некоторые атрибуты (свойства) и помеченный именем.

Способ хранения файлов на дисках компьютера называется **файловой системой**.

Папка – это специальное место на диске, где содержится информация о файлах и других папках.

Окно – один из основных элементов интерфейса Windows. В окнах запускаются все программы Windows. В виде окон открываются папки, можно открыть сразу несколько окон.

В любой момент времени можно работать только с одним окном. Окно, с которым непосредственно работает пользователь, называется **активным**. Активное окно располагается поверх других окон и его заголовок выделен более ярким цветом.

Окно папки имеет следующую стандартную структуру:

Строка заголовка – название папки (имя программы).

Системный значок – находится в верхнем левом углу окна, при щелчке откроется служебное меню (команды, позволяющие управлять размером и расположением окна).

Строка меню – список команд. При щелчке на каждом из пунктов этого меню открывается «ниспадающее» меню, пункты которого позволяют проводить операции с содержимым окна или с окном в целом.

Панель инструментов – содержит командные кнопки для выполнения наиболее часто встречающихся операций.

Адресная строка – показывает путь доступа к текущей папке, позволяет выполнить быстрый переход к другим разделам файловой структуры с помощью раскрывающейся кнопки справа.

Рабочая область – в окнах папок отображает значки объектов, хранящихся в папке; в окнах программ отображает активный документ; в окне документа отображает содержимое документа.

Полосы прокрутки – если количество объектов слишком велико (или размер окна слишком мал) по правому и нижнему краям рабочей области могут отображаться полосы прокрутки, с помощью которых можно «прокручивать» содержимое папки в рабочей области.

Строка состояния – содержит дополнительную информацию.





3. Работа с файловой структурой при помощи программы Проводник


Иерархическая структура, в виде которой операционная система отображает файлы и папки диска, называется **файловой структурой**.

Просмотреть файловую структуру можно при помощи программы **Проводник**.

Программа **Проводник** является основным инструментом для файловых операций в системе Windows. Она отображает содержимое папок, позволяет копировать, перемещать, удалять, переименовывать папки и файлы, запускать программы.

Рабочая область окна Проводника состоит из двух областей: области дерева папок и устройств (слева) и области активной (открытой) папки (справа). В любой момент времени только одна папка может быть открыта.

Папка может быть **развернута** или **свернута**. Если в папке имеются вложенные папки, то рядом с ней появляется узел, отмеченный знаком . Щелчок на узле разворачивает папку (т.е. показывает структуру вложенных папок), знак узла  меняется на знак . Щелчок на значке узла  сворачивает папку.

Папка может быть **раскрыта** или **закрыта**. Если щелкнуть на значке папки, то она раскроется  и её содержимое отобразится в области активной папки Проводника. Закрывается папка при раскрытии любой другой папки.

Задание на лабораторную работу:

1. Откройте папку на Рабочем столе папку **Мои документы**.

2. Создайте две новые папки:

Первую папку при помощи строки меню окна **Файл→Создать→Папку**;

Вторую папку с помощью контекстного меню (правая кнопка мыши): выберите пункт **Создать→Папку**.

3. Переименуйте папки:

С помощью строки меню окна: выделите одну из папок и выполните команду **Файл→Переименовать**. Дайте имя **Папка 1**.

С помощью контекстного меню (правая кнопка мыши): выделите другую папку и выберите команду **Переименовать**. Дайте имя **Папка 2**.

4. Откройте обе папки одним из способов:

Двойной щелчок;

С помощью контекстного меню командой **Открыть**;

С помощью строки меню окна: **Файл → Открыть**;

С помощью клавиши **ENTER** при выделенном значке папки.

5. Расположите окна папок рядом, не перекрывая друг друга (если необходимо, измените размер окна). Чтобы каждая папка открывалась в отдельном окне, выполните команду: **Пуск → Панель управления → Свойства папки → вкладка Вид → Дополнительные параметры** и установите флажок **Открывать каждую папку в отдельном окне**.

6. В первой папке создайте 4 файла: два текстовых документа и два рисунка.

Вызовите контекстное меню и выберите команду **Создать** → **Текстовый документ** или **Создать** → **Точечный рисунок**.

7. Все файлы из **Папки 1** переместите в **Папку 2** разными способами:

С помощью строки меню: в окне **Папки 1** выделите **Файл 1** и выполните команду **Правка** → **Вырезать**. Перейдите в окно **Папки 2** и выполните команду **Правка** → **Вставить**.

С помощью панели инструментов: выделите **Файл 2** и нажмите кнопку **«Вырезать»** (ножницы). Перейдите в окно **Папки 2** и нажмите кнопку **«Вставить»**. Если панель инструментов не содержит этих кнопок, добавьте их при помощи команды **Вид** → **Панели инструментов** → **Настройка**, выделите необходимые имеющиеся кнопки и выполните команду **Добавить**.

С помощью контекстного меню: выделите **Файл 3** и нажмите правую кнопку мыши для вызова контекстного меню, выберите команду **Вырезать**. Перейдите в окно **Папки 2** и на пустом месте вызовите контекстное меню, выберите команду **Вставить**.

С помощью комбинации клавиш на клавиатуре: выделите **Файл 4**, одновременно нажмите клавиши **Ctrl** и **X**. Перейдите в окно **Папки 2** и нажмите одновременно **Ctrl** и **V**.

8. Все файлы из **Папки 2** скопировать в **Папку 1** разными способами:

С помощью строки меню: в **Папке 2** выделите **Файл 1** и выполните команду **Правка** → **Копировать**. Перейдите в окно **Папки 1** и выполните команду **Правка** → **Вставить**.

С помощью панели инструментов: в **Папке 2** выделите **Файл 2** и нажмите кнопку **«Копировать»**, перейдите в окно **Папки 1** и нажмите кнопку **«Вставить»**.

С помощью контекстного меню: в **Папке 2** выделите **Файл 3** и нажмите правую кнопку мыши для вызова контекстного меню, выберите команду **Копировать**. Перейдите в окно **Папки 1** и на пустом месте вызовите контекстное меню и выберите команду **Вставить**.

С помощью комбинации клавиш на клавиатуре: в **Папке 2** выделите **Файл 4** и нажмите одновременно **Ctrl** и **C**. Перейдите в окно **Папки 1** и нажмите **Ctrl** и **V**.

9. Из **Папки 1** удалите все файлы разными способами:

С помощью строки меню: выделите **Файл 1** и выполните команду **Файл** → **Удалить**.

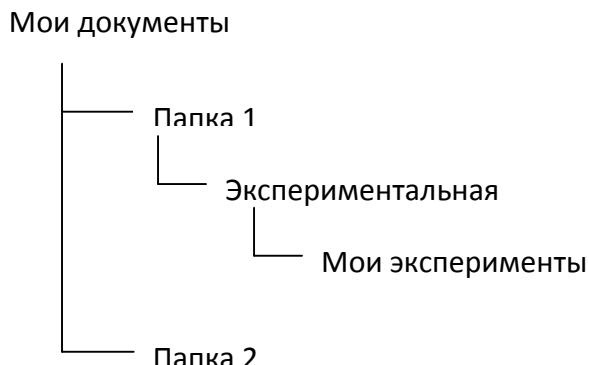
С помощью панели инструментов: выделите **Файл 2** и нажмите кнопку **«Удалить»**.

С помощью контекстного меню: выделите **Файл 3** и нажмите правую кнопку мыши для вызова контекстного меню, выберите команду **Удалить**.

С помощью клавиши на клавиатуре: выделите **Файл 4** и нажмите клавишу **Delete**.

10. Закройте окна **Папка 1** и **Папка 2**.

11. Запустите программу **Проводник** при помощи кнопки **Пуск**→**Все программы**→**Стандартные**→**Проводник** (или контекстного меню кнопки **Пуск**) и создайте следующую структуру папок:



2. Найдите папку **Мои документы** в области дерева папок, в которой должны находиться Ваши папки: **Папка 1** и **Папка 2**.

3. Откройте **Папку 1** и в области активной папки создайте новую папку одним из известных Вам способов и дайте ей имя **Экспериментальная**. В области дерева папок убедитесь в том, что рядом со значком папки **Папка 1** образовался знак узла **+**, свидетельствующий о том, что папка имеет вложенные папки. При нажатии на знак узла **+** происходит разворачивание **Папки 1** и он меняется на знак **-**.

4. Откройте папку **Экспериментальная** в области дерева папок Проводника.

В области активной папки не должно отображаться никакого содержимого, поскольку эта папка пуста.

5. Создайте в области активной папки Проводника новую папку **Мои эксперименты** внутри папки **Экспериментальная**. Убедитесь в том, что в области дерева папок рядом со значком папки **Экспериментальная** образовался знак узла **+**. Разверните узел и рассмотрите образовавшуюся структуру вложенных папок в области дерева папок Проводника.

6. Откройте **Папку 2**, убедитесь в том, что в области активной папки в ней есть содержимое (созданные ранее Вами файлы), но в области дерева папок рядом со значком папки нет знака узла **+**, так как в ней нет вложенных папок.

7. Покажите результаты работы преподавателю.

8. Удалите **Папку 1** и **Папку 2** в **Корзину** одним из известных Вам способом.

9. Очистите **Корзину** при помощи контекстного меню.

Лабораторная работа № 2.

ОСНОВЫ РАБОТЫ В MS WORD. СОЗДАНИЕ ТЕКСТОВЫХ ДОКУМЕНТОВ.

Цель работы: изучить основы создания и редактирования текстовых документов в текстовом процессоре MS Word, научиться создавать таблицы, создавать формулы с помощью средства Microsoft Equation.

Microsoft Word – текстовый процессор, предназначенный для создания, просмотра и редактирования текстовых документов, с локальным применением простейших форм таблично-матричных алгоритмов. Выпускается корпорацией Microsoft в составе пакета Microsoft Office.

Текст вводится с помощью алфавитно-цифровых клавиш. Место документа, в которое происходит ввод текста (точка ввода), отмечается на экране вертикальной чертой, которую называют **курсором**.

Для ввода прописных букв используется клавиша **Shift**.

Для фиксации режима ввода заглавных букв служит клавиша **CapsLock**.

Клавиша **Backspace (←)** (слева от курсора) и **Delete** (справа от курсора) удаляют символы или выделенные фрагменты текста.

Для переключения с русского на латинский алфавит необходимо использовать комбинации клавиш **Ctrl+Shift** или **Alt+Shift**.

Таблица 1 – Выделение текста с помощью мыши

| Для выделения... | Выполнить действие... |
|---------------------------------|---|
| Слова | Двойной щелчок мыши по требуемому слову. |
| Предложения | Нажать клавишу Ctrl и одновременно одинарный щелчок левой кнопкой мыши в любом месте предложения. |
| Строчного фрагмента | При нажатой левой кнопки мыши протаскиванием курсора по тексту. |
| Абзаца | а) Установить указатель мыши у выделяемого абзаца слева, вне пределов абзаца. Дважды нажать левую кнопку мыши. б) Тройной щелчок левой кнопкой мыши в любом месте абзаца. |
| Прямоугольного фрагмента | При нажатых клавише Alt и левой кнопки мыши протаскивание курсора по тексту. |
| Фрагмента произвольного размера | Установить указатель мыши в начале фрагмента. Нажать левую кнопку мыши и, не отпуская её, переместить указатель в конец необходимого фрагмента. Для завершения выделения отпустить кнопку мыши. |

Для выделения всего текста используют команду строки меню **Правка → Выделить всё**.

Задание на лабораторную работу

1. Откройте программу **Microsoft Word** через ярлык на Рабочем столе либо через Главное меню **Пуск → Все программы → Microsoft Word**.
2. Наберите текст.

Предмет и основные понятия информатики

Информатика – это комплексная, техническая наука, которая систематизирует приемы создания, сохранения, воспроизведения, обработки и передачи данных средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ими.

3. Проверьте правописание при помощи команды **Рецензирование → Правописание**.

!!Обратите внимание! Перед выполнением команд форматирования текст необходимо выделить с помощью курсора мыши.

4. Установите параметры форматирования текста:

Заголовок:

При помощи команды строки меню **Формат → Шрифт** вкладка **Шрифт** установите: шрифт – **Arial**; начертание – **Полужирный**; размер – **14**.

При помощи команды строки меню **Формат → Абзац** вкладка **Отступы и интервалы** установите: выравнивание – **По центру**; интервал перед и после – **Авто**.

Текст:

При помощи панели инструментов **Форматирование** установите: шрифт – **Times New Roman**; размер шрифта – **12**; выравнивание – **По ширине**, начертание определите по образцу.

При помощи команды строки меню **Формат → Абзац** вкладка **Отступы и интервалы** установите: в разделе **Отступ** → **Первая строка** → **Отступ** на 1,3 см; в разделе **Интервал** → **Междустрочный** – полусторонний.

5. Сохраните текст, для этого выполните команду строки меню **Файл → Сохранить как..**, дайте имя файлу и нажмите кнопку **Сохранить**.

6. Перейдите на следующую страницу при помощи одновременного нажатия клавиш **Ctrl** и **Enter**.

7. С помощью команды строки меню **Таблица → Вставить → Таблица** выберите число строк и столбцов по предложенному образцу.

Коэффициенты перевода производных единиц от байта

| | Байт | Килобайт | Мегабайт | Гигабайт |
|-----------|----------|-------------|-------------|-------------|
| б | 1 | $2^{(-10)}$ | $2^{(-20)}$ | $2^{(-30)}$ |
| Кб | 2^{10} | 1 | $2^{(-10)}$ | $2^{(-20)}$ |
| Мб | 2^{20} | 2^{10} | 1 | $2^{(-10)}$ |
| Гб | 2^{30} | 2^{20} | 2^{10} | 1 |

8. Выделите таблицу и примените стиль оформления с помощью команды строки меню **Таблица** → **Автоформат**.

9. Сохраните файл при помощи команды строки меню **Файл** → **Сохранить**.

10. На третьей странице вставьте формулу при помощи команды строки меню **Вставка** → **Объект** → **Microsoft Equation**

$$S = \sum_{x=1}^n x^2.$$

11. На этой же странице вставьте следующие формулы:

$$\int \frac{dx}{\sqrt{a^2 - x^2}} = \arcsin \frac{x}{a} + C = -\arccos \frac{x}{a} + C_1$$

$$\sum_{n=1}^{\infty} \frac{n(2i-1)^n}{3^n}$$

$$\lim_{x \rightarrow \infty} \frac{\sqrt{x}}{\sqrt{x + \sqrt{x + \sqrt{x}}}}$$

$$y = \frac{(\alpha \sin \beta x - \beta \cos \beta x)e^{\alpha x}}{\alpha^2 + \beta^2}$$

12. Сохраните файл при помощи команды строки меню **Файл** → **Сохранить**.

Лабораторная работа № 3.

ОСНОВЫ РАБОТЫ В MS WORD. ФОРМАТИРОВАНИЕ ТЕКСТОВЫХ ДОКУМЕНТОВ.

Цель работы – научиться оформлять текстовые документы, созданные в MS Word в соответствии заданными требованиями.

Форматирование текста — комплекс действий, направленных на создание правильного оформления текста соответственно тем или иным нормам.

Задание на лабораторную работу:

1. Откройте программу **Microsoft Word** через ярлык на Рабочем столе либо через Главное меню **Пуск → Все программы → Microsoft Word**.
2. Установите поля страницы:
 - верхнее 2,5 см;
 - нижнее 2,5 см;
 - левое 3 см;
 - правое 1 см.
3. Сохраните созданный Вами документ на рабочем диске в своей папке под именем Фамилия_Работа_2.docx.
4. Напечатайте на второй странице следующий текст:

Текстовый редактор MS Word

Большую популярность среди текстовых редакторов разных производителей получило приложение Word компании Microsoft. Вместе с приложениями Excel, PowerPoint, Access, OneNote, Outlook, Publisher текстовый редактор Word входит в состав пакета MS Office.

Microsoft Word предлагает пользователям широкий инструментарий для подготовки документов различного назначения: макетов книг, брошюр, отчетов, деловых писем, бланков и пр. С момента выхода первых версий MS Word претерпел большие изменения как в пользовательском интерфейсе, так и в функциональном наполнении. Так, начиная с выпуска 2007 г. в приложении введен новый ленточный интерфейс, изменивший подход к расположению меню и команд. Последние версии приложения позволяют удаленно работать над одним документом группам пользователей, надежно защищать документы от несанкционированного доступа с помощью шифрования и электронной цифровой подписи, а также предоставляют пользователям «облачный» сервис.

5. Для заголовка текста установите размер шрифта 16 пт, начертание полужирное, выравнивание абзаца по центру, интервал после абзаца 12 пт.

6. Для основного текста установите размер шрифта 14 пт, выравнивание по ширине, отступ первой строки каждого абзаца 1,25 см, множитель 1,4 междустрочного интервала, интервал между абзацами 6 пт.
7. Установите режим автоматической расстановки переносов.
8. Проверьте правописание во всем тексте.
9. Используя буфер обмена, скопируйте напечатанный Вами текст с заголовком так, чтобы он повторился в документе четыре раза.
10. После каждой копии текста вставьте разрыв страницы.
11. На второй странице документа оформите заголовок текста как объект WordArt.
12. На третьей странице документа разбейте текст на две колонки. Заголовок текста должен располагаться в первой колонке и быть выровненным по центру.
13. На четвертой странице документа разбейте текст на три колонки. Заголовок текста должен располагаться над колонками и выровнен по центру абзаца. Установите вертикальные линии - разделители колонок.
14. На пятой странице документа первый абзац заключите в рамку из двойных линий, примените заливку текста желтым цветом, установите синий цвет символов. Вставьте любой рисунок из коллекции картинок MS Office. Установите размер рисунка 4 x 4 см без сохранения пропорций, обтекание текстом

Вокруг рамки. Рисунок должен располагаться слева от второго абзаца.

15. Сохраните файл.

16. На шестой странице создайте схему «Структура информатики» по предложенному образцу, используя панель инструментов **Рисование**, если такой панели нет в окне **MS WORD**, выполните команду строки меню **Вид**

→ **Панели инструментов** и установите флажок ☒ Рисование.

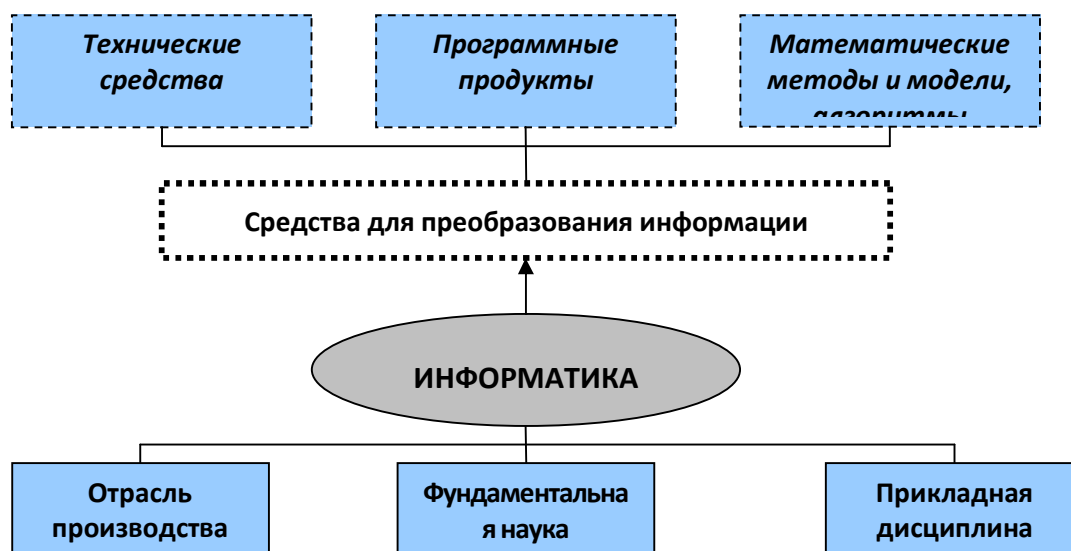


Рис 1. Структура информатики

17. Сохраните файл, покажите результаты работы преподавателю.

Лабораторная работа № 4. ОСНОВЫ РАБОТЫ В MS EXCEL. СОЗДАНИЕ И ФОРМАТИРОВАНИЕ ЭЛЕКТРОННЫХ ТАБЛИЦ.

Цель работы: изучить основы работы в табличном процессоре Excel, научиться создавать табличные файлы, содержащие данные, и применять к ним форматирование в соответствии с заданными требованиями.

Приложение Excel входит в состав всех выпусков пакета Microsoft Office и предназначено для работы с электронными таблицами. Специалисты считают MS Excel лидером среди программных средств, позволяющих создавать и обрабатывать электронные таблицы. За почти тридцать лет своего существования (первая версия приложения была выпущена в 1985 году) Excel получил значительное расширение функциональных возможностей, а его внешний вид не раз перевоплощался. В настоящей версии интерфейс построен с использованием ленты и вкладок команд, оформленных в едином стиле с другими программами пакета Microsoft Office.

Возможности электронных таблиц Excel очень широки. К их числу относятся: решение вычислительных задач, визуализация данных с помощью диаграмм, обработка и анализ статистических данных, экономическое моделирование, подготовка отчетов, организация хранилищ (баз) данных и их обработка и многие другие.

Электронная таблица (ЭТ) – это двумерный массив, состоящий из столбцов и строк. Столбцы идентифицированы буквами латинского алфавита (**A, B, C...**), строки идентифицированы арабскими цифрами (**1, 2, 3...**). Место пересечения столбца и строки называется ячейкой, например, **A3**. Ячейка, в которой находится табличный курсор, является активной. В ЭТ можно работать как с одной ячейкой, так и с группой ячеек, которые образуют блок или диапазон. Блок идентифицируется именами левой верхней и правой нижней ячеек. Имена ячеек в блоке разделяются двоеточием (:), например **A1:A4**.

Для выполнения вычислений Excel предлагает специальные средства расчетов – формулы и функции. Формула в программе Excel начинается со знака (=); в формуле не должно быть пробелов; при вводе формулы используются символы латинского алфавита.

Ссылкой (адресом) определяется ячейка или группа ячеек, используемых в формуле. Для указания ссылки на ячейку вводится буква заголовка столбца и номер строки. Адресация может быть **абсолютной** и **относительной**. По умолчанию ссылки на ячейки в формулах рассматриваются как относительные и при копировании формул адреса в ссылках автоматически изменяются. При абсолютной адресации адреса ссылок при копировании не изменяются. Для изменения способа адресации используют клавишу **F4**. Элементы номера ячейки, использующие абсолютную адресацию, предваряются символом **\$**.

Задание на лабораторную работу:

1. Запустите MS Excel. Введите следующие данные:

| | A | B | C | D | E | F |
|----|-------------|-----------|------|-----------|------------|---|
| 1 | Поступление | Категория | Цена | Поставщик | Реализация | |
| 2 | Январь | Шоколад | 56 | Ланта | Март | |
| 3 | Январь | Шоколад | 89 | Ланта | Март | |
| 4 | Январь | Шоколад | 23 | Парус | Апрель | |
| 5 | Январь | Шоколад | 120 | Парус | Апрель | |
| 6 | Январь | Кофе | 320 | Парус | Март | |
| 7 | Январь | Кофе | 265 | Парус | Март | |
| 8 | Январь | Печенье | 35 | Парус | Апрель | |
| 9 | Январь | Печенье | 35 | Марс | Апрель | |
| 10 | Январь | Печенье | 35 | Марс | Март | |
| 11 | | | | | | |

2. Отредактируйте заголовки колонок: Категория измените на Товар, Цена измените на Цена, р.

3. Разместите между строками с информацией о шоколаде и кофе две пустых строки и введите в них данные (диапазон A6:E7):

| | | | | |
|---------|-----|----|-------|-----|
| Февраль | Сок | 55 | Ланта | Май |
| Март | Сок | 55 | Парус | Май |

4. Вставьте между колонками Цена и Поставщик колонку Количество и заполните ее данными:

| Количество |
|------------|
| 230 |
| 560 |
| 320 |
| 280 |
| 244 |
| 488 |
| 300 |
| 200 |
| 576 |
| 288 |
| 350 |

5. Разместите колонку Поставщик после колонки Товар.

6. Дополните таблицу (диапазон A13:F16) следующей информацией:

| | | | | | |
|---------|---------|-------|-----|-----|--------|
| Февраль | Шоколад | Ланта | 85 | 200 | Апрель |
| Февраль | Сок | Парус | 45 | 200 | Май |
| Февраль | Кофе | Марс | 400 | 200 | Июнь |
| Февраль | Печенье | Марс | 48 | 200 | Июль |

7. Вставьте перед колонкой Поступление пустую колонку и введите заголовок № п/п.
8. Используя маркер заполнения, пронумеруйте строки таблицы цифрами от 1 до 15 в колонке № п/п.
9. Удалите из таблицы строку под номером 4 в колонке № п/п и исправьте нумерацию строк в данной колонке.
10. Используя команду Главная → Редактирование → Найти и выделить → Заменить, в колонке Поставщик замените Ланта на Лавита.
11. Разместите над заголовками колонок две пустые строки и введите в ячейку A1 название таблицы: Реализация товаров со склада № 22.
12. Используя команду Главная → Выравнивание → Объединить и поместить в центре, разместите заголовок по центру колонок.
13. В ячейку A2 введите слово Дата, в ячейку B2 введите текущую дату, в ячейку E2 введите слово Время, в ячейку F2 введите текущее время.
14. Нарисуйте границы в таблице.
15. Сравните созданную Вами таблицу с таблицей, представленной ниже. При наличии расхождений внесите исправления.

| | A | B | C | D | E | F | G |
|----|---|-------------|---------|-----------|---------|------------|------------|
| 1 | Реализация товаров со склада №22 | | | | | | |
| 2 | Дата | 23.02.2014 | | | Время | 11:36 | |
| 3 | № п/п | Поступление | Товар | Поставщик | Цена, р | Количество | Реализация |
| 4 | 1 | Январь | Шоколад | Лавита | 56 | 230 | Март |
| 5 | 2 | Январь | Шоколад | Лавита | 89 | 560 | Март |
| 6 | 3 | Январь | Шоколад | Парус | 23 | 320 | Апрель |
| 7 | 4 | Февраль | Сок | Лавита | 55 | 244 | Май |
| 8 | 5 | Март | Сок | Парус | 55 | 488 | Май |
| 9 | 6 | Январь | Кофе | Парус | 320 | 300 | Март |
| 10 | 7 | Январь | Кофе | Парус | 265 | 200 | Март |
| 11 | 8 | Январь | Печенье | Парус | 35 | 576 | Апрель |
| 12 | 9 | Январь | Печенье | Марс | 35 | 288 | Апрель |
| 13 | 10 | Январь | Печенье | Марс | 35 | 350 | Март |
| 14 | 11 | Февраль | Шоколад | Лавита | 85 | 200 | Апрель |
| 15 | 12 | Февраль | Сок | Парус | 45 | 200 | Май |
| 16 | 13 | Февраль | Кофе | Марс | 400 | 200 | Июнь |
| 17 | 14 | Февраль | Печенье | Марс | 48 | 200 | Июль |

Лабораторная работа № 5. ОСНОВЫ РАБОТЫ В MS EXCEL. ВЫЧИСЛЕНИЯ В MS EXCEL.

Цель работы: научиться решать математические задачи в табличном процессоре Excel, научиться создавать формулы с помощью встроенных математических функций

Формулы в Excel позволяют проводить сколь угодно сложные вычисления и получать результаты моментально.

Особенности написания формул:

- Каждая формула начинается со знака «=».
- Участвовать в вычислениях могут значения из ячеек и функции.
- В качестве привычных нам математических знаков операций используются операторы.
- При вставке записи в ячейке по умолчанию отражается результат вычислений.
- Посмотреть конструкцию можно в строке над таблицей..

Excel предлагает большое количество функций, которые разбиты по категориям. Посмотреть полный перечень можно, нажав на кнопку Fx около строки формул или открыв раздел «Формулы» на панели инструментов.

Самые распространенные функции Excel:

| Категория | Мат. запись | Запись Excel | Описание | Число аргументов |
|----------------|-------------|-------------------|--|------------------|
| Математические | $ x $ | Abs(A2) | Абсолютная величина | 1 |
| | Sin x | Sin(A2) | Синус | 1 |
| | Cos x | Cos(A2) | Косинус | 1 |
| | tg x | Tan(A2) | Тангенс | 1 |
| | Arcsin x | Asin(A2) | Арксинус | 1 |
| | Ln x | Ln(A2) | Натуральный логарифм | 1 |
| | e^x | Exp(A2) | Экспонента | 1 |
| | \sqrt{x} | Корень(A2) | Квадратный корень | 1 |
| | Σ | Сумм(A1:D5;f1:k7) | Сумма | любое |
| Статистические | | МАКС(A1:A110) | Возвращает наибольшее значение из набора. | Любое |
| | | МИН(AB23:AC1015) | Возвращает наименьшее значение из набора. | Любое |
| | | СРЗНАЧ (A1:A11) | Среднее арифметическое аргументов | Любое |
| | | ДИСПР(C1:C100) | Вычисляет дисперсию генеральной совокупности | 30 |
| Дата, время | | СЕГОДНЯ() | Возвращает текущую дату и | нет |

Задание на лабораторную работу:

1. Откройте программу **Microsoft Excel** через ярлык на Рабочем столе, либо через Главное меню **Пуск** → **Все программы** → **Microsoft Excel**, откроется новая рабочая книга, состоящая из нескольких рабочих листов и при помощи команды меню **Файл** → **Сохранить как...**, сохраните рабочую книгу под своим именем.

2. Определите значения выражений для $x=3$ и $y=5$

а) $\frac{1+x^2}{3y}$ (Ответ 0,666667)

б) $-2y^2 + \frac{x^3}{4x+5}$ (Ответ -48,4118)

в) $\frac{x + \frac{2+y}{x^2}}{y + \frac{1}{x^2+10}}$ (Ответ 0,747685)

3. Вычислите значения квадратов и кубов первых 10 чисел. Решение оформите в виде таблицы.

| x | x^2 | x^3 |
|-----|-------|-------|
| 1 | | |
| 2 | | |
| 3 | | |
| ... | ... | |

4. Заполните данными таблицу и выполните вычисление в ней.

| | | | |
|----------------------|-------|-----------------|-----------------|
| Цена 1 литра бензина | АИ-95 | 34,9 | |
| | АИ-92 | 32,8 | |
| | | | |
| | Литры | Стоимость АИ-95 | Стоимость АИ-92 |
| | 5 | | |
| | 10 | | |
| | 15 | | |
| | 20 | | |
| | 25 | | |
| | 30 | | |
| | 35 | | |
| | 40 | | |

5. Заполните исходными данными таблицу. Вычислите площади прямоугольников по заданным ширине и длине сторон.

| | | Длина | | | | | | | | |
|--------|----|-------|---|---|---|---|---|---|---|----|
| | | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Ширина | 2 | | | | | | | | | |
| | 3 | | | | | | | | | |
| | 4 | | | | | | | | | |
| | 5 | | | | | | | | | |
| | 6 | | | | | | | | | |
| | 7 | | | | | | | | | |
| | 8 | | | | | | | | | |
| | 9 | | | | | | | | | |
| | 10 | | | | | | | | | |

6. Определите значения выражений

а) $\sqrt[3]{723}$ (Ответ 8,97524059)

б) $\log_2 3$ (Ответ 1,584962501)

в) $\sqrt{\sin x + 2}$ для $x = 4$ (Ответ 1,11498767)

7. Заполните данными таблицу и выполните вычисления в ней. В тригонометрических функциях аргумент задается в радианах.

| х (градусы) | х (радианы) | $\sin x$ | $\cos x$ | $\sqrt{\sin^2 x + 1}$ |
|----------------|----------------|----------|----------|-----------------------|
| 0 | | | | |
| 15 | | | | |
| 30 | | | | |
| 45 | | | | |
| 60 | | | | |
| 75 | | | | |
| 90 | | | | |
| 105 | | | | |
| 120 | | | | |
| 135 | | | | |
| 150 | | | | |
| 165 | | | | |
| 180 | | | | |

Лабораторная работа № 6. ОСНОВЫ РАБОТЫ В MS EXCEL. СОЗДАНИЕ ГРАФИКОВ В MS EXCEL.

Цель работы: научиться визуализировать табличные данные с помощью графиков средствами MS Excel.

Диаграммы и графики в MS Excel служат для графического отображения данных, что более наглядно с точки зрения пользователя. С помощью диаграмм удобно наблюдать за динамикой изменений значений исследуемых величин, проводить сравнения различных данных, представление графической зависимости одних величин от других. Чтение и оценка большого количества данных, которые визуализированы с помощью графиков и диаграмм значительно упрощается. MS Excel располагает эффективным многофункциональным инструментом для этой визуализации, благодаря которому можно построить диаграммы и графики разнообразных типов и назначений.

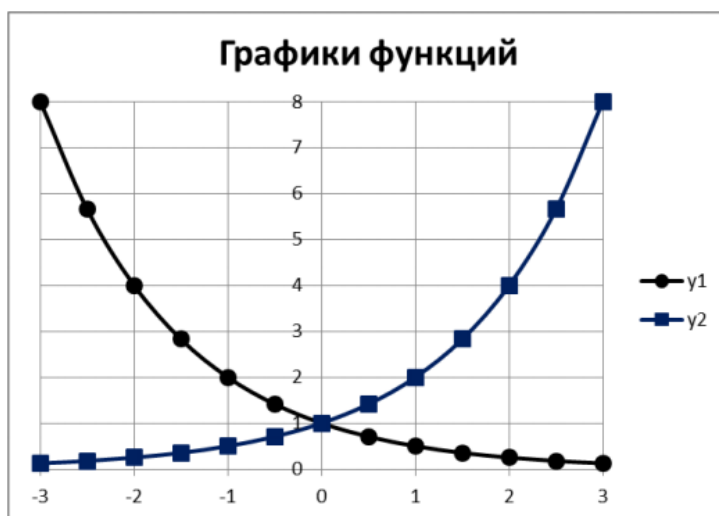
Задание на лабораторную работу:

1. Запустите MS Excel.
2. Создайте на Листе 1 таблицу для построения графиков функций

$y_1 = \frac{1}{2^x}$ $y_2 = 2^x$ на отрезке $[-3; 3]$ с шагом 0,5.

| x | -3 | -2,5 | -2 | -1,5 | -1 | -0,5 | 0 | 0,5 | 1 | 1,5 | 2 | 2,5 | 3 |
|----|-------|--------|------|--------|-----|--------|---|--------|-----|--------|------|--------|-------|
| y1 | 8 | 5,6569 | 4 | 2,8284 | 2 | 1,4142 | 1 | 0,7071 | 0,5 | 0,3536 | 0,25 | 0,1768 | 0,125 |
| y2 | 0,125 | 0,1768 | 0,25 | 0,3536 | 0,5 | 0,7071 | 1 | 1,4142 | 2 | 2,8284 | 4 | 5,6569 | 8 |

2. Ниже таблицы вставьте диаграмму. Тип диаграммы – точечная с гладкими кривыми и маркерами.
3. Установите цвета линий графика и маркеров: для y_1 – черный, для y_2 – темно-синий. Измените тип маркеров на графиках.
4. Добавьте название диаграммы. Отобразите вертикальные и горизонтальные линии сетки.
5. Установите отображение значений горизонтальной оси на отрезке от -3 до 3 .
6. Сравните построенную Вами диаграмму с представленной справа. При наличии расхождений между ними внесите в Вашу диаграмму исправления.



7. Введите данные на лист 2.

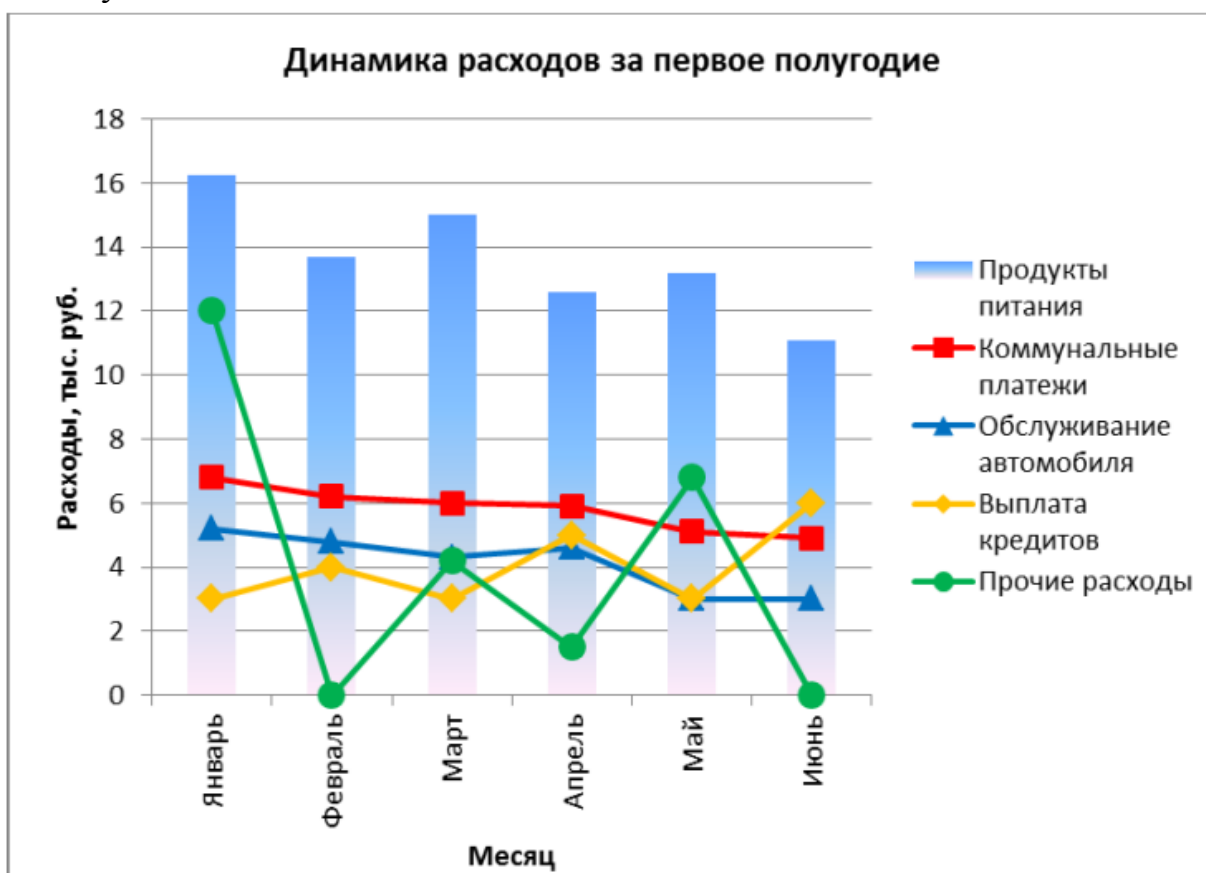
| Расходы за первое полугодие (тыс. руб.) | | | | | | |
|---|--------|---------|------|--------|------|------|
| | Январь | Февраль | Март | Апрель | Май | Июнь |
| Продукты питания | 16,25 | 13,7 | 15 | 12,6 | 13,2 | 11,1 |
| Коммунальные платежи | 6,8 | 6,2 | 6 | 5,9 | 5,1 | 4,9 |
| Обслуживание автомобиля | 5,2 | 4,8 | 4,3 | 4,6 | 3 | 3 |
| Выплата кредитов | 3 | 4 | 3 | 5 | 3 | 6 |
| Прочие расходы | 12 | 0 | 4 | 4,2 | 1,5 | 6,8 |

8. На Листе 2 ниже таблицы постройте диаграмму график с маркерами.

9. Увеличьте размер диаграммы.

10. Измените для ряда Продукты питания тип диаграммы на гистограмму с группировкой.

11. Установите для гистограммы ряда Продукты питания градиентную заливку «Рассвет».



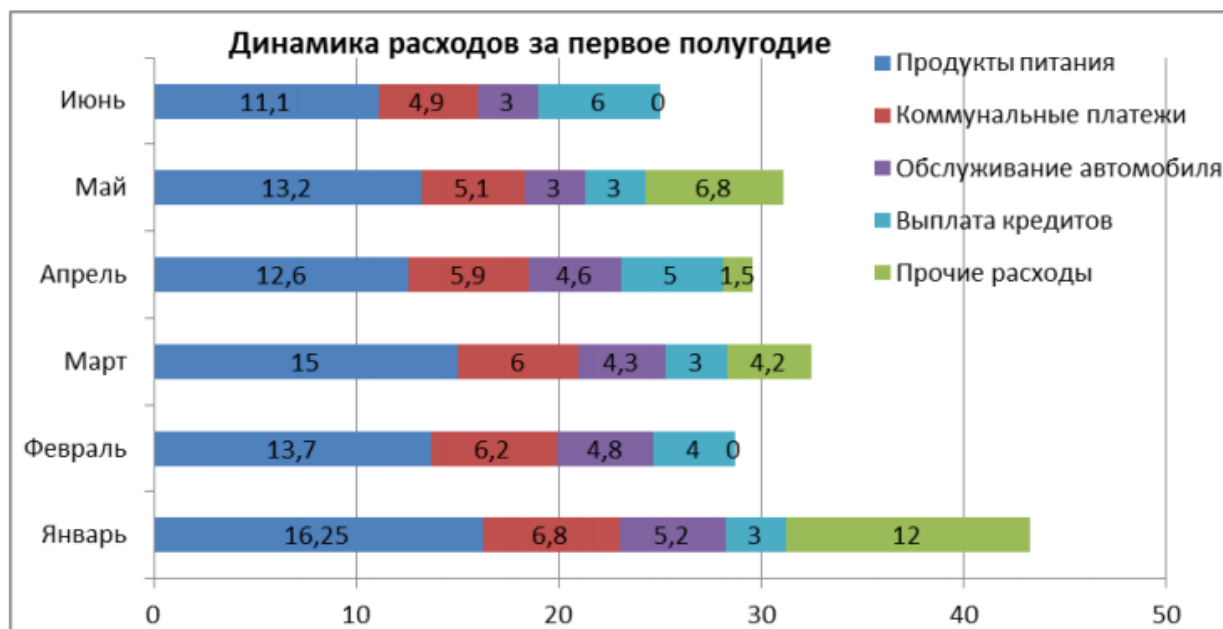
7. Установите для линий графика следующие цвета: коммунальные платежи – красный, обслуживание автомобиля – синий, выплата кредитов – оранжевый, прочие расходы – зеленый.

8. Вставьте название диаграммы «Динамика расходов за первое полугодие».

9. Установите вертикальное выравнивание подписей на горизонтальной оси категорий.

10. Сравните построенную Вами диаграмму с представленной. При наличии расхождений между ними внесите в Вашу диаграмму необходимые изменения.

11. На этом же рабочем листе для исходных данных постройте линейчатую диаграмму с накоплениями.
12. Установите размеры диаграммы: высота – 8 см., ширина – 20 см.
13. Вставьте название диаграммы и подписи данных.
14. Сравните построенную Вами диаграмму с представленной. При наличии расхождений между ними внесите в Вашу диаграмму необходимые изменения.



15. Сохраните файл. Покажите результаты работы преподавателю.

Лабораторная работа № 7.

ОСНОВЫ РАБОТЫ В MS ACCESS. СОЗДАНИЕ БАЗЫ ДАННЫХ.

Цель работы: изучить основные принципы работы с программой MS Access, научиться проектировать и создавать простые базы данных, изучить особенности построения связей между таблицами.

MS Access – это система управления базами данных (СУБД) корпорации Microsoft, в которой данные представляются в виде реляционной модели (табличном виде).

Основные компоненты MS Access:

- таблица – компонент приложения, предназначенный для записи и хранения данных;
- запрос – элемент, предназначенный для получения информации из одной или нескольких таблиц, является средством для обращения к связанным БД и стороннему ПО;
- форма – объект используется для представления введенной информации в более удобном для пользователя виде;
- отчет позволяет получить конечный результат в виде готового документа;
- макрос представляет собой элемент, содержащий в себе последовательное описание для выполнения того или иного действия, с помощью него можно задать команду, которая будет выполнять определенную задачу, например, проверка изменения данных в одной из таблиц;
- модуль – компонент, который содержит в себе программное обеспечение, написанное на языке программирования Visual Basic, С его помощью редактор существенно расширяет функционал. Достигается это благодаря использованию функций и процедур, призванных реагировать на те или иные изменения;
- страница доступа с помощью которой можно получить доступ к удаленным базам, хранящимся на других персональных компьютерах.

Основным объектом базы данных является таблица, которая состоит из записей (строк) и полей (столбцов). На пересечении записи и поля образуется ячейка, в которой содержатся данные. Каждому полю таблицы присваивается уникальное имя, которое не может содержать более 64 символов. В каждом поле содержатся данные одного типа.

Основные типы данных:

| Тип | Описание |
|--|--|
| Текстовый Поле МЕМО | Используется для хранения символьных или числовых данных, не требующих вычислений. В свойстве <i>Размер поля</i> задается максимальное количество символов, которые могут быть введены в данное поле. По умолчанию размер устанавливается в 50 знаков. Максимальное количество символов, которые могут содержаться в текстовом поле, – 255 Предназначено для ввода текстовой информации, по объему пре- |
| Числовой Дата/Время | вышающей 255 символов; может содержать до 65 536 символов Предназначен для хранения числовых данных, используемых в математических расчетах. На вкладках <i>Общие</i> и <i>Подстановка</i> можно установить свойства числового поля, среди которых <i>Размер поля</i> , <i>Формат поля</i> , <i>Число десятичных знаков</i> Используется для представления даты и времени. Выбор конкретного формата даты или времени устанавливается в |
| Денежный | свойстве <i>Формат даты</i> Предназначен для хранения данных, точность представления |
| Счетчик Логический | которых колеблется от 1 до 4 знаков после запятой. Целая часть может содержать до 15 десятичных знаков Предназначен для автоматической вставки уникальных последовательных (увеличивающихся на 1) или случайных чисел в качестве номера новой записи. Номер, присвоенный записи, не может быть удален или изменен. Поля с этим типом данных используются в качестве ключевых полей таблицы Предназначен |
| Поле объекта OLE | для хранения одного из двух значений, интерпретируемых как «Да / Нет», «Истина / Ложь», «Вкл. / Выкл.» Содержит данные, созданные в других программах, которые используют протокол OLE. Это могут быть, например, документы Word, электронные таблицы Excel, рисунки, звуковые и видеозаписи и др. Объекты OLE связываются с базой данных |

Задание на лабораторную работу:

1. Запустите Microsoft Access.
2. Создадим базу данных «Фирма». Сотрудники данной организации работают с клиентами и выполняют их заказы.

Если все сведения поместить в одной таблице, то она станет очень неудобной для работы. В ней начнутся повторы данных. Всякий раз, когда сотрудник Иванов будет работать с какой-либо фирмой, придется прописывать данные о сотруднике и клиенте заново, в результате чего можно допустить множество ошибок. Чтобы уменьшить число ошибок, можно исходную таблицу разбить на несколько таблиц и установить связи между ними. Это будет более рационально, чем прежде.

Таким образом, необходимо создать 3 таблицы: Сотрудники, Клиенты и Заказы.

Сотрудники

| Имя поля | Тип данных |
|------------------|-------------|
| Код сотрудника | Счетчик |
| Фамилия | Текстовый |
| Имя | Текстовый |
| Отчество | Текстовый |
| Должность | Текстовый |
| Телефон | Текстовый |
| Адрес | Текстовый |
| Дата рождения | Дата/Время |
| Заработная плата | Денежный |
| Фото | Объект OLE |
| Эл_почта | Гиперссылка |

Клиенты

| Имя поля | Тип данных |
|-------------------------|-------------|
| Код клиента | Счетчик |
| Название компании | Текстовый |
| Адрес | Текстовый |
| Номер телефона | Текстовый |
| Факс | Числовой |
| Адрес электронной почты | Гиперссылка |
| Заметки | Поле МЕМО |

Заказы

| Имя поля | Тип данных |
|----------------------|------------|
| Код заказа | Счетчик |
| Код клиента | Числовой |
| Код сотрудника | Числовой |
| Дата размещения | Дата/Время |
| Дата исполнения | Дата/Время |
| Сумма | Денежный |
| Отметка о выполнении | Логический |

3. Отдельные таблицы, содержащие информацию по определенной теме, необходимо связать в единую структуру базы данных. Для связывания таблиц следует задать ключевые поля. Ключ состоит из одного или нескольких полей, значения которых однозначно определяют каждую запись в таблице. Наиболее подходящим в качестве ключевого поля является «Счетчик», так как значения в данном поле являются уникальными (т. е. исключают повторы).

4. Откройте таблицу Сотрудники в режиме Конструктора.
5. Нажмите правой кнопкой мыши на поле Код сотрудника и в появившемся контекстном меню выберите команду Ключевое поле. Если в таблице

необходимо установить несколько ключевых полей, то выделить их можно, удерживая клавишу Ctrl.

6. Выйдите из режима Конструктора, предварительно сохранив таблицу под именем «Друзья»; ключевые поля не задавайте.

7. Откройте таблицу «Друзья» двойным щелчком мыши и заполните в ней 10 строк.

8. Добавьте поля «Отчество» и «Дата рождения», для этого:

1) установите курсор на поле, перед которым нужно вставить новый столбец;

2) выполните команду: вкладка ленты Режим таблицы → панель инструментов Поля и столбцы → Вставить;

3) щелкнув два раза на Поле1, переименуйте его в «Отчество», а Поле2 – «Дата рождения».

9. Перейдите в режим Конструктора командой: вкладка ленты Главная → Режимы → Конструктор.

10. Для поля «Дата рождения» установите тип данных Дата / время; в свойствах поля выберите Краткий формат даты.

11. Отформатируйте таблицу следующим образом:

1) цвет сетки – темно-красный;

2) цвет фона – голубой;

3) цвет текста – темно-красный, размер – 12 пт, начертание – курсив.

12. Переименуйте поле «Хобби» в «Увлечения».

13. Удалите запись под номером 8.

14. Измените размер ячеек так, чтобы были видны все данные. Для этого достаточно два раза щелкнуть левой кнопкой мыши на границе полей.

15. Расположите поля в следующем порядке: «№», «Фамилия», «Имя», «Отчество», «Телефон», «Дата рождения», «Увлечения», «Адрес», «Индекс», «Фото», «Эл_почта».

16. Заполните пустые ячейки таблицы.

17. В режиме Конструктора добавьте поле «Семейное положение», в котором будет содержаться фиксированный набор значений – замужем, не замужем, женат, не женат. Для создания раскрывающегося списка будем использовать Мастер подстановок:

1) установите тип данных Мастер подстановок; 2) в появившемся диалоговом окне выберите строку «Будет введен фиксированный набор значений» и нажмите кнопку Далее; 3) число столбцов – 1; 4) введите данные списка – замужем, не замужем, женат, не женат; 5) нажмите кнопку Готово.

18. С помощью раскрывающегося списка заполните новый столбец. Поскольку таблица получилась широкая, то при заполнении данного столбца возникают некоторые неудобства: не видно фамилии человека, для которого заполняется поле «Семейное положение». Чтобы фамилия была постоянно видна при заполнении таблицы, необходимо воспользоваться командой Закрепить столбцы из контекстного меню поля «Фамилия».

19. Покажите работу преподавателю.

Лабораторная работа № 8. СОЗДАНИЕ ПРЕЗЕНТАЦИЙ В MS POWER POINT.


Цель работы: создать и просмотреть презентацию на тему «Предмет и основные понятия информатики» при помощи программы PowerPoint.

Презентация (от лат. praesentatio) — это общественное представление чего-либо нового, недавно появившегося, созданного, например: книги, журнала, кинофильма, телепрограммы, организации. Презентация представляет собой сочетание компьютерной анимации, графики, видео, музыки и звукового ряда, которые организованы в единую среду. Как правило, презентация имеет сюжет, сценарий и структуру, организованную для удобного восприятия информации. Их можно представлять в электронном виде и распространять через Интернет.

Microsoft PowerPoint (полное название — Microsoft Office PowerPoint) — это программа для создания и проведения презентаций, позволяющая придать необходимый эффектный внешний вид. Является частью Microsoft Office и доступна в редакциях для операционной системы Microsoft Windows.

Задание на лабораторную работу:

1. Запустите программу **PowerPoint** (**Пуск** → **Все программы** → **MS PowerPoint**). Сохраните презентацию под именем «**Информатика**» при помощи команды строки меню **Файл** → **Сохранить как...**
2. В области задач (справа) выберите команду **Создание** → **Из мастера автосодержания**, если области задач нет, то воспользуйтесь командой строки меню **Файл** → **Создать**.
3. В окне **Мастер автосодержания** щёлкните на кнопке **Далее**.
4. В следующем окне **Вид презентации** выберите строку **Общий доклад** и щёлкните на кнопке **Далее**.
5. На этапе выбора **стиля презентации** установите переключатель **Презентация на экране** и щёлкните на кнопке **Далее**.
6. На этапе выбора **параметров презентации** введите в окно заголовка «**Информатика**», в окне нижнего колонтитула повторите название. Щёлкните на кнопке **Далее**, а потом **Готово**.
7. На открывшемся рабочем поле в окне **структуры** (слева) выделите поочередно слайды с **5** по **9** при помощи кнопки **Shift** и удалите их (**Delete**). В презентации должно остаться четыре слайда.
8. Перейдите к первому слайду при помощи полосы прокрутки и в окне слайда в блоке под заголовком замените имя, указанное по умолчанию собственным именем в качестве автора разработки. Задайте выравнивание текста по центру при помощи панели инструментов и поместите блок ниже красной линии методом перетаскивания.

9. Затем выделите блок с заголовком и поместите его в верхней части слайда при помощи перетаскивания при появлении двунаправленной стрелки  (выше красной линии).

10. Вставьте в слайд рисунок при помощи команды строки меню **Вставка** → **Рисунок** → **Картинки**. В области задач выберите **Коллекция картинок** (внизу справа) и в окне **Избранное** выберите в **Коллекции Microsoft Office** в разделе **Бизнес** картинку **Вычислительная техника**. Скопируйте картинку при помощи команд строки меню **Правка** → **Копировать** и **Правка** → **Вставить**.

11. Удалите с титульного слайда колонтитул при помощи команды строки меню **Вид** → **Колонтитулы**, установите флажок на пункте **Не показывать на титульном слайде** и нажмите кнопку **Применить ко всем**.

12. Перейдите на второй слайд и вместо заголовка «**Введение**» наберите текст «**Задачи информатики**».

13. Перейдите в блок маркированного списка и введите вместо шаблонного текста следующее содержание:

- Исследует информационные процессы в социальных системах;
- Разрабатывает информационную технику;
- Создает новейшие технологии преобразования информации.

14. Выводите текстовые блоки на втором слайде по своему усмотрению.


15. Перейдите к третьему слайду и введите заголовок «**Единицы информации**». Удалите нижний текстовый блок (он нам здесь не нужен) и при помощи команды меню **Вставка** → **Таблица** в окне **Вставка таблицы** задайте число столбцов – **2** и число строк – **3**, нажмите **ОК**.

16. Заполните таблицу:

| | |
|----------------|---------------------------------|
| 1 Кбайт | 2^{10} байт |
| 1 Мбайт | 2^{20} байт |
| 1 Гбайт | 2^{30} байт |

Верхний индекс вводится при помощи команды строки меню **Формат** → **Шрифт**, установите флажок на пункте **Надстрочный**.

17. Перейдите к четвёртому слайду. В поле заголовка введите текст: «**Структура информатики**». Удалите второй текстовый блок.

18. При помощи команды **Вставка** → **Организационная диаграмма** в окне **Библиотека диаграмм** выберите первый тип диаграммы и нажмите **ОК**. В первую рамку в диаграмме введите текст **Информатика**. В остальные рамки введите **Отрасль производства**, **Фундаментальная наука** и **Прикладная дисциплина**. Стил диаграммы можно выбрать при помощи команды **Автоформат**  на панели инструментов **Организационная диаграмма**. Увеличьте размер шрифта на своё усмотрение.

19. Перейдите к первому слайду, выделите заголовок и выберите эффект анимации при помощи команды меню **Показ слайдов** → **Настройка анимации** в области задач нажмите на кнопке **Добавить эффект**. В появившемся подменю выберите эффекты: **Вход** → **Жалюзи**; **Выделение** → **Вращение**; **Выход** → **Шашки**; **Пути перемещения** → **Вправо-вниз**.

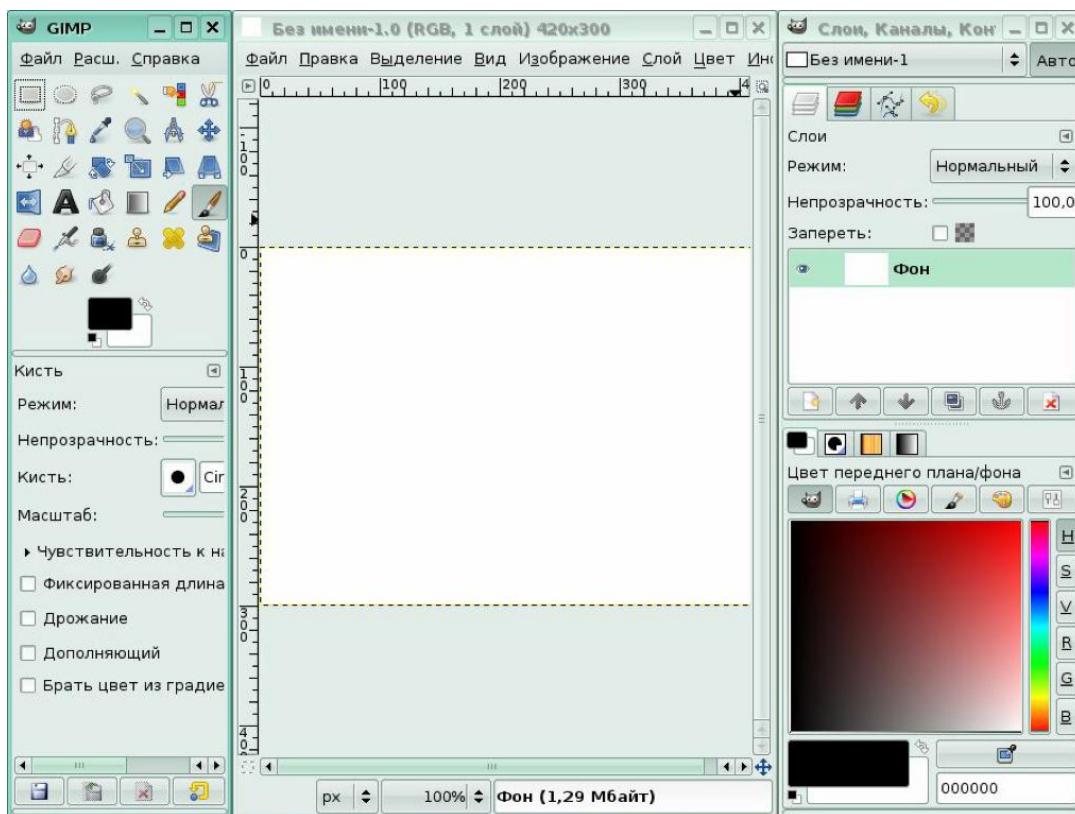
20. Выполните команду строки меню **Показ слайдов**→**Смена слайдов**. В пункте **Смена слайдов** снимите флажок **По щелчку** и установите его на **Автоматически после** и задайте время **2 секунды**. Нажмите на кнопке **Применить ко всем слайдам**.
21. Примените эффекты анимации для других слайдов во всех блоках.
22. Выполните команду **Показ слайдов** → **Эффекты анимации** в области задач (справа) в пункте **Дизайн слайда** выберите **Шаблоны оформления**. Выберите шаблон оформления по своему усмотрению.
23. Вернитесь к первому слайду, запустите показ презентации при помощи команды меню **Показ слайдов** → **Начать показ**.
24. Сохраните изменения в презентации при помощи команды строки меню **Файл** → **Сохранить**.

Лабораторная работа № 9. ИЗУЧЕНИЕ ГРАФИЧЕСКИХ РЕДАКТОРОВ.

Цель работы: Получить базовые навыки работы в графическом редакторе GIMP.

GNU Image Manipulation Program или GIMP («Гимп») — свободно распространяемый растровый графический редактор, программа для создания и обработки растровой графики и частичной поддержкой работы с векторной графикой. Проект основан в 1995 году Спенсером Кимбеллом и Питером Маттисом как дипломный, в настоящий момент поддерживается группой добровольцев.

Типичные задачи, которые можно решать при помощи GIMP, включают в себя создание графики и логотипов, масштабирование и кадрирование фотографий, раскраску, комбинирование изображений с использованием слоёв, ретуширование и преобразование изображений в различные форматы.



В GIMP присутствует набор инструментов цветокоррекции:

- кривые;
- уровни;
- микшер каналов;
- постеризация;
- тон-насыщенность;
- баланс цветов;

- яркость-контраст;
- обесцвечивание.

При помощи фильтров, инструментов, масок и слоёв с разными типами наложения (всего 22) можно:

- выравнивать заваленный горизонт;
- убирать искажения, вносимые оптикой;
- корректировать перспективу;
- выполнять клонирование объектов с учётом перспективы;
- кадрировать фотографии;
- удалять дефекты вроде пыли на матрице (штамп, лечебная кисть);
- имитировать использование различных цветофильтров;
- «вытаскивать» потерянную детализацию в тенях;

Задание на лабораторную работу

1. Получить эффект старинной фотографии. Для этого:
 - Открыть любое изображение и перевести в режим RGB (Изображение – Режим – RGB).
 - Изменить цвет на коричневый командой Цвет – Тонировать. Если необходимо, усилить контрастность и яркость. Это можно сделать в этом же диалоге окне, либо командой Цвет – Яркость/Контраст.
2. Выделить цветом различные части изображения. Для этого:
 - Перевести в режим цветного изображения RGB (Изображение – Режим – RGB).
 - Откорректировать контрастность и яркость всего изображения (Цвет–Яркость/Контраст).
 - Выделить фигурной рамкой ("Lasso (Свободное выделение)") один объект и изменить его цвет командой Цвет – Тонировать в сине-фиолетовый.
 - Выбрать обратное выделение командой Выделение – Инвертировать и изменить цвет окружения на красно-коричневый (тонирование).
3. Создать черно-белую графическую иллюстрацию для одноцветной печати (например, шелкографии). Для этого:
 - Откорректировать яркость, перевести в черно-белую графику командой Цвет – Обесцвечивание. В появившемся диалоге окне выбрать основы оттенков – среднее.
 - Выбрать центральную (наиболее интересную) часть снимка прямоугольной рамкой.
 - Командой Выделение – Инвертировать изменить выделенную область на поля и применить команду Цвет – Инвертировать.
 - При необходимости кадрировать полученную картинку.
4. Получить иллюстрацию для фотоальбома. Для этого:
 - Перевести изображение в цветной режим RGB.
 - Выбрать прямоугольной рамкой центральную часть изображения.

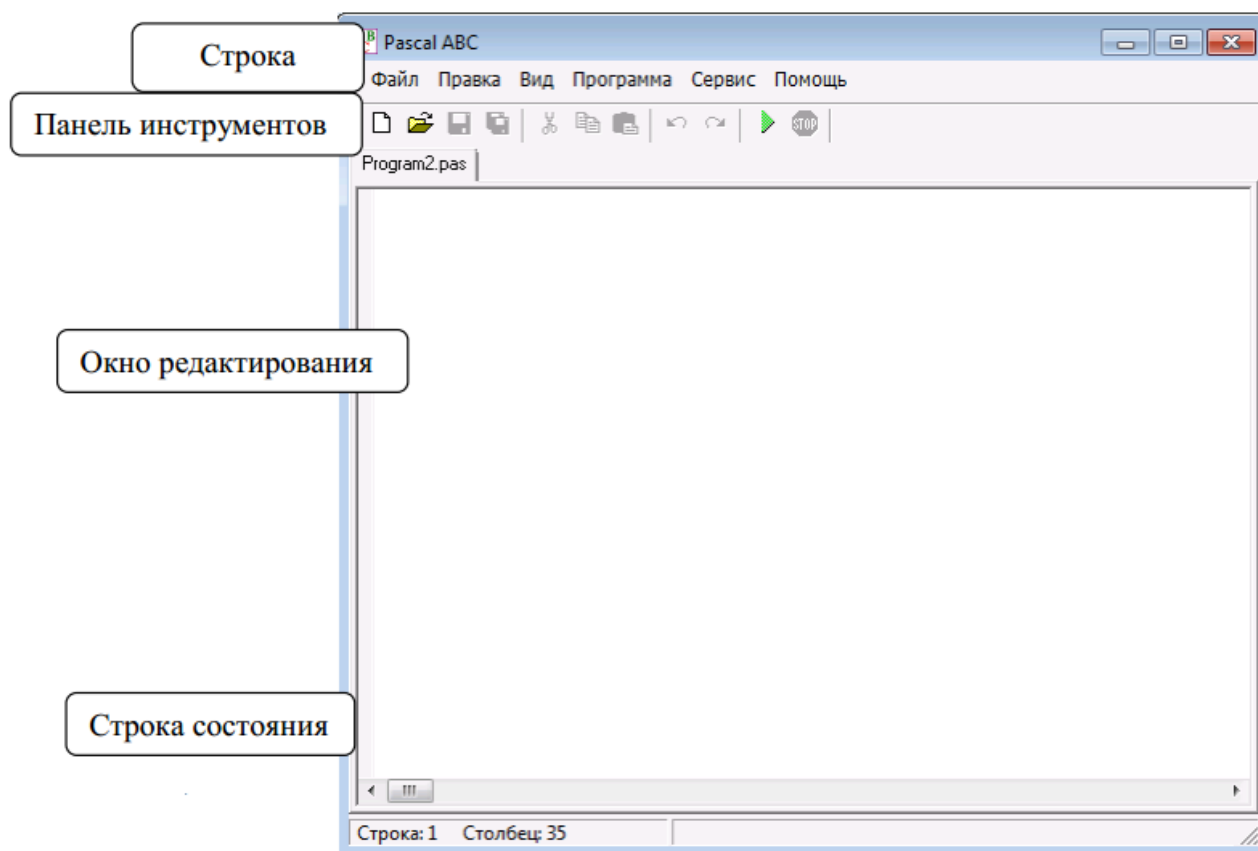
- Применить тонирование выделенной области в серо-зеленый.
 - Изменить выбранную область командой Выделение – Инвертировать и перевести поля в черно-белую графику командой Цвет – Обесцвечивание.
 - Повторить команду Выделение – Инвертировать и обвести выделенную область рамкой белого цвета шириной 2-6 пикселей командой Правка– Обвести выделенное.
5. Имитировать вид через влажное стекло. Для этого:
- Перевести изображение в цветной режим RGB.
 - Тонировать в красно-коричневый (под старую фотографию).
 - Выбрать произвольную часть изображения инструментом "Свободное выделение".
 - Изменить область выделения командой Выделение – Инвертировать.
 - Применить к выделенной области фильтр Фильтр – Размывание – Гауссово размывание, с произвольными настройками.
 - Повторить команду Выделение – Инвертировать, увеличить контрастность и уменьшить яркость.
6. Имитировать вид через замерзшее стекло. Для этого:
- Перевести изображение в цветной режим RGB, тонировать полностью в серо-синий или серо-фиолетовый.
 - Выбрать произвольную часть изображения.
 - Изменить область выделения Выделение – Инвертировать.
 - Увеличить яркость и уменьшить контрастность выделенной области и применить эффект Фильтр/Шум/Рассеивание.
 - Повторить команду Выделение – Инвертировать, увеличить контрастность и уменьшить яркость выделенной области.
7. Создать на базе черно-белой фотографии цветную иллюстрацию для двухцветной печати (типа шелкографии). Для этого:
- Перевести изображение в цветной режим RGB, а затем – в черно-белую графику командой Цвет – Обесцвечивание.
 - Командой Цвет – Карта – Замена цвета, заменить белый на светло-оранжевый, а черный – на сине-фиолетовый.
 - Кадрировать полученное изображение.
8. Создать на базе черно-белой фотографии графическую иллюстрацию для цветного рекламного буклета. Для этого:
- Перевести изображение в цветной режим RGB.
 - Перевести тональное изображение в 4-6 основных оттенков серого командой Цвет – Постеризация.
 - Командой Цвет – Карта – Замена цвета, заменить белый на светло-желтый, нейтральные – на красно-коричневые, черный –на сине-фиолетовый.
 - При необходимости кадрировать полученное изображение.
9. Показать полученные изображения преподавателю.

Лабораторная работа № 10.

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ.

Цель работы: изучить интерфейс и основы работы в среде программирования Pascal ABC, познакомиться со структурой программы и основными операторами языка программирования Паскаль.

Среда программирования Pascal ABC – это пакет взаимосвязанных файлов, которые позволяют набирать, редактировать, запускать и отлаживать программы.



Первая строка экрана – меню интегрированной среды, следующая строка – панель инструментов, нижняя строка экрана – строка состояния интегрированной среды. Между ними расположено окно редактирования – рабочее поле, в котором можно открывать несколько вкладок для разных программ. Окно редактирования предназначено для ввода и редактирования текста программы. Место ввода информации обозначено курсором. В верхней левой части окна редактирования выводится служебное имя редактируемого файла, например: Program1.pas.

Структура программы

Программа на языке Pascal ABC имеет следующий вид:

program имя программы;

раздел описаний

begin

операторы

end.

Первая строка называется заголовком программы и не является обязательной. Раздел описаний предназначен для объявления всех встречающихся в программе данных и их характеристик. Этот раздел в свою очередь содержит: объявление меток, констант, типов, переменных, процедур и функций. После каждого описания ставится символ ";". Раздел описаний может отсутствовать. Операторы – команды, из которых состоит программа на Паскале и многих других языках. Операторы отделяются один от другого символом ";".

Основные операторы вывода:

Write('комментарий', <список переменных>);

Writeln('комментарий', <список переменных>);

Правила записи и выполнения оператора WriteLn те же, что и у Write, с одним исключением – после выполнения оператора Write следующий оператор Write или WriteLn печатает свою информацию в той же строке, а после выполнения оператора WriteLn – с начала следующей.

Основные операторы ввода

Readln(<список переменных>);

Read(<список переменных>);

Оператор read используется крайне редко, в основном используется оператор readln. После выполнения оператора readln программа останавливается и ждет ввода данных с клавиатуры. Пользователь набирает желаемые значения переменных, разделяя их пробелами (или Enter) и нажимает Enter. Оператор интерпретирует введенные символы, переводя их во внутреннее (машинное) представление соответствующих переменных.

Запуск и остановка программы

Для запуска программы в текущем окне редактора следует нажать клавишу F9 или кнопку панели инструментов. Программа вначале компилируется во внутреннее представление, после чего, если не найдены ошибки, программа начинает выполняться. При выполнении программы кнопка запуска программы становится неактивной, кнопка остановка программы, наоборот, активной и в строке статуса отображается информация "Программа выполняется". Выполнение программы можно в любой момент прервать нажатием комбинации клавиш Ctrl-F2. При этом в окне вывода появится сообщение **Программа прервана пользователем.**

Арифметические операции

| Арифметическая операция | Описание операции | Пример |
|-------------------------|---|------------------------|
| + | Сложение | $5+2=7$ |
| - | Вычитание | $5-2=3$ |
| * | Умножение | $5*2=10$ |
| / | Деление | $5.0/2.0 = 2.5$ |
| div | Целочисленное деление (деление без остатка) | $5 \text{ div } 2 = 2$ |
| mod | Остаток от деления | $5 \text{ mod } 2 = 1$ |

Стандартные математические функции

| Функция | Результат функции | Пример |
|------------|---|-----------------------------------|
| abs(x) | Модуль числа (абсолютное значение) | abs(-2) = 2 |
| sqr(x) | Квадрат числа | sqr(2) = 4 |
| sqrt(x) | Квадратный корень числа | sqrt(4) = 2.0 |
| pi | Число пи ~ 3.14... | |
| random(x) | Случайное число из диапазона $0 \leq \dots < x$ | random(75)=52 |
| sin(x) | Синус числа, x должен задаваться в радианах | sin(pi) = 0 |
| cos(x) | Косинус числа, x должен задаваться в радианах | cos(0) ~ 1.0 |
| arctan(x) | Арктангенс числа | arctan(0)=0 |
| ln(x) | Логарифм натуральный, $x > 0$ | ln(1) = 0 |
| exp(x) | Экспонента (число e в степени x) $e=2.718281\dots$ | exp(1) ~ 2.72 |
| power(x,y) | Степень числа | Power(2,3) = 8 |
| int(x) | Целая часть числа | int(3.6) = 3.0 |
| frac(x) | Дробная часть числа | frac(5.3) = 0.3 |
| trunc(x) | Целое число без округления | trunc(3.6) = 3 |
| round(x) | Целое число с округлением | round(3.6) = 4 |
| odd(x) | Значение функции true, если x нечетен, и false в противном случае | odd(75) = True odd(76) = False |

Переменные

Исходные данные и результаты в программировании называют величинами.

Величины, которые меняются в процессе работы программы, называют переменными, а те, которые не меняются, — константами.

Переменная характеризуется именем, типом и значением. Имя переменной может содержать только буквы латинского алфавита, цифры и символ подчеркивания (_). Имя не может начинаться с цифры. Никакие другие символы в записи имен не разрешаются. В качестве имени нельзя использовать служебные слова языка PASCAL, например, такие как begin и end.

Основные типы переменных

| Тип | Описатель типа (имя) | Значение типа (пример) |
|------------------------|----------------------|------------------------|
| Действительный | real | 16.32 |
| | | -1.6320000000E+01 |
| Целый | integer | 30000 |
| Длинный целый | longint | 2000000000 |
| Символьный (литерный) | char | 'a' |
| Строковый | string | 'задача' |
| Логический (булевский) | boolean | False |
| | | true |
| Интервальный | M1..M2 | 100 |

Описание переменных (VAR)

Любая переменная, встречающаяся в программе, должна быть описана в разделе объявления переменных.

Описание начинается со служебного слова VAR и имеет следующую форму записи:

VAR имя переменной : тип ;

В одном разделе допускается описание нескольких переменных, например:

VAR

B :INTEGER; { переменная целого типа }

SUM :REAL; { действительного типа }

K :CHAR; { символьного типа }

LOG :BOOLEAN; { логического типа }

Если несколько переменных имеют одинаковый тип, то их можно объединить в список. Под списком понимается последовательность элементов (в нашем случае – переменных), разделенных запятой.

Задание на лабораторную работу:

1. Используя стандартные функции и правила записи арифметических выражений, вычислить значение выражения $4(x-3)^6 - 7(x-3)^3 + 2$, если $x=1$, вывести результат.
2. Самостоятельно вычислить значения выражений и вывести результат.

| |
|--|
| $\frac{a^3+b^3}{ab}$, если $a=3, b=5$ |
| $\frac{-b+\sqrt{b^2+4ab}}{2a}$, если $a=4, b=7$ |
| $\frac{x}{1+\frac{x^2}{3+(2x)^3}}$, если $x=5$ |
| $\sqrt{a^2 + b^2 - 2ab \cos y}$, если $a=23, b=75, y=1$ |
| $\sin x \cos y + \cos x \sin y$, если $x=90^\circ, y=180^\circ$ |

3. Написать и выполнить программу на компьютере: программа вводит одно число и вычисляет его квадрат, вычитая из него удвоенное значение введенного числа.
4. Самостоятельно написать и выполнить программы на компьютере:
 1. Программа вводит два числа, выводит сумму квадратов этих чисел минус их произведение.
 2. Программа вводит два числа, выводит сумму их модулей.

Лабораторная работа № 11. ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ.

Цель работы: повторить этапы решения задач на компьютере, освоить составление алгоритма решения разветвляющихся задач на языке блок-схем, познакомиться с условным оператором языка программирования Паскаль, научиться записывать алгоритм решения разветвляющихся программ на язык программирования Паскаль.

Перед написанием любой программы обычно разрабатывается алгоритм, который представляется в виде блок-схемы.

Основные элементы для построения блок-схем

| Элемент блок-схемы | Наименование | Содержание |
|---|--------------------------------|--|
|  | Начало (конец) | Начало или конец алгоритма |
|  | Блок ввода-вывода данных | Общее обозначение ввода (вывода) данных |
|  | Блок вычислений | Вычислительные действия или последовательность действий |
|  | Логический блок (блок условия) | Выбор направления выполнения алгоритма в зависимости от некоторого условия |
|  | Блок модификации | Модификация – организация циклических конструкций (начало цикла) |

Линейные алгоритмы описывают решение задач с последовательным выполнением действий. Обычно такие действия идут в следующем порядке:

- 1) ввод исходных данных (может отсутствовать, тогда данные задаются внутри программы),
- 2) последовательные команды – обычно вычислительного характера;
- 3) вывод результатов (должен присутствовать обязательно).

Разветвляющаяся программа – это программа, в котором в зависимости от выполнения или невыполнения некоторого условия выполняется либо одна, либо другая последовательность действий.

Условный оператор (оператор if)

Условный оператор может записываться в полной и неполной форме.

Полная форма условного оператора

IF логическое выражение **THEN** оператор 1

ELSE оператор 2

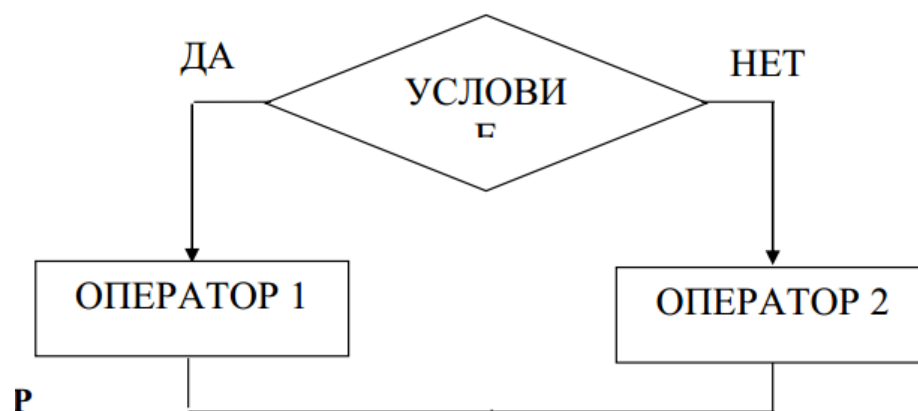
Здесь IF (если), THEN (тогда), ELSE (иначе) – служебные слова, оператор 1, оператор 2 – простые или составные операторы. Если логическое выражение истинно, тогда выполняется оператор 1, иначе (если логическое выражение ложно) – оператор 2.

Логическое выражение (условие) в операторе IF..THEN...ELSE может быть простым или сложным.

Простое условие состоит из левой части, оператора сравнения и правой части.

| Оператор | Операция |
|----------|------------------|
| < | Меньше |
| <= | Меньше или равно |
| > | Больше |
| >= | Больше или равно |
| = | Равно |
| <> | Не равно |

Блок-схема полного условного оператора



Задание на лабораторную работу:

1. Нарисовать блок-схему алгоритма и составить программу, чтобы компьютер по закону Ома для участка цепи, запрашивая с клавиатуры значения напряжения на концах участка и его сопротивления, определял и выводил на экран значение силы тока ($I=U/R$).
2. Нарисовать блок-схему алгоритма и написать программу, вычисляющую и выводящую на экран число бит, если задано число Мбайт. Количество Мбайт запрашивайте у пользователя программы.
3. Нарисовать блок-схему алгоритма и составить программу, позволяющую вычислить объем пирамиды при известных площади основания и высоте. Результаты вывести с 3 знаками после десятичной точки. ($V=(1/3)*S*h$)
4. Нарисовать блок-схему алгоритма и составить программу, анализирующую данные пожарного датчика в помещении, которая выводит сообщение «Пожароопасная ситуация», если температура в комнате превысила 60°C . Значение температуры вводится с клавиатуры.
5. Нарисовать блок-схему алгоритма и составить программу, проверяющую, что введенное число является четным или нечетным.
6. Нарисовать блок-схему алгоритма и составить программу для решения квадратного уравнения ($ax^2 + bx + c = 0$). Коэффициенты a , b , c вводятся с клавиатуры.

Лабораторная работа № 12.

ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ.

Цель работы: повторить этапы решения задач на компьютере, освоить составление алгоритма решения циклических задач на языке блок-схем, познакомиться с операторами цикла языка программирования Паскаль, научиться записывать алгоритм решения циклических программ на язык программирования Паскаль.

Алгоритм называется циклическим, если он содержит многократное выполнение одних и тех же операторов при различных значениях промежуточных данных. Число повторений этих операторов может быть задано в явной (цикл с известным заранее числом повторений) или неявной (цикл с неизвестным заранее числом повторений) форме.

Операторы цикла используются для вычислений, повторяющихся многократно. В Паскале три вида циклов:

- цикл с параметром `for` (арифметический цикл),
- цикл с предусловием `while`,
- цикл с постусловием `repeat`.

Каждый из них состоит из определенной последовательности операторов. Блок, ради выполнения которого и организуется цикл, называется телом цикла.

Остальные операторы служат для управления процессом повторения вычислений: это начальные установки, проверка условия продолжения цикла и модификация параметра цикла.

Оператор цикла с параметром FOR

Оператор цикла с параметром применяют тогда, когда заранее известно число повторений одной и той же последовательности операторов. Начальные и конечные значения параметра цикла могут быть представлены константами, переменными или арифметическими выражениями.

```
FOR K:=M1 TO M2 DO  
BEGIN  
оператор1;  
.....  
оператор n;  
34  
END;
```

Цикл с предусловием WHILE

Цикл с предусловием `WHILE` используется тогда, когда число повторений оператора цикла заранее не известно, а задается некоторое условие продолжения цикла.

WHILE <логическое выражение / условие> DO
BEGIN
Операторы циклической части программы
END

Сначала выполняется значение условия. Пока оно истинно, выполняются операторы циклической части. Когда только оно становится ложным, происходит выход из цикла. Если условие ложно, то цикл не выполняется ни разу.

Оператор цикла с постусловием Repeat

Этот оператор отличается от цикла с предусловием WHILE тем, что проверка условия производится после очередного выполнения тела цикла. Это обеспечивает выполнение цикла хотя бы один раз.

REPEAT
Операторы циклической части программы
UNTIL<логическое выражение>

Операторы циклической части выполняются повторно до тех пор, пока значение логического выражения ложно. Условием прекращения циклических вычислений является истинное значение логического выражения.

Задание на лабораторную работу

1. Нарисовать блок-схему алгоритма и написать программу, определяющую значения функции $y = 5x^2 - 3x + 3$, для значений x , изменяющихся от -8 до 2 с шагом 1. Решить задачу, используя:
 - а) цикл с параметром,
 - б) цикл с предусловием,
 - в) цикл с постусловием.
2. Нарисовать блок-схему алгоритма и написать программу, запрашивающую имя пользователя и печатающую "Привет, Имя!" 10 раз.
3. Нарисовать блок-схему алгоритма и написать программу выводящую сумму чисел от 1 до n , где n – вводится пользователем с клавиатуры.
4. Нарисовать блок-схему алгоритма и написать программу, выводящую сумму чисел от 1 до n , где n – вводится пользователем с клавиатуры.
5. Нарисовать блок-схему алгоритма и написать программу, выводящую все чётные числа, начиная с числа N и до числа M . Числа N и M задает пользователь.

Лабораторная работа № 13. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ОДНОМЕРНЫХ МАССИВОВ.

Цель работы: научиться объявлять, заполнять одномерные массивы и выполнять действия с элементами массивов.

Массив (array, от французского – сплошной) – упорядоченная группа фиксированного количества переменных одного типа, имеющая общее имя. Каждый элемент такой совокупности обозначается именем массива с индексом (номером элемента в массиве).

Элемент массива – отдельная переменная, входящая в массив. **Индекс элемента массива** – номер элемента в массиве. Особенность массивов заключается в том, что все элементы массива являются данными одного типа. Массиву присваивается имя, при помощи которого можно ссылаться на него как на единое целое, так и на любой из его элементов.

Обращение к элементу массива:

<имя массива>[<индекс>],

Требования к имени массива:

- не должен содержать символов русского алфавита;
- не должен начинаться с цифры;
- не должен содержать знак пробела.

Индекс (номер элемента) может быть выражением порядкового типа.

Индекс 1 2 3 N

Имена элементов записываются следующим образом:

A[1] A[2] A[3] A[n]

A - имя массива, 1-N - индекс массива.

В зависимости от количества используемых индексов, массивы могут быть одномерные (строка), двумерные (таблица), трехмерные и т.д.

Каждое из значений, составляющих массив, называется его компонентой (или элементом массива). Все элементы массива имеют один и тот же тип. Это очень важное замечание. Именно поэтому массивы относятся к однородным типам данных, в отличие от записей, объединений, деревьев, файлов и прочих типов данных, элементы которых могут быть различных типов.

Тип элементов называется базовым типом массива. Размер массива в Паскале не может быть более 65 520 байт.

Необходимо различать понятия размер массива и размерность массива.

Размер массива - количество элементов в нём (диапазон изменения индексов).

Размерность массива - число индексов, определяющих положение элемента в массиве.

Описание массива:

```
Type t=array[1..n] of тип;  
Var a:t;  
Var a: array[1..n] of тип;
```

Заполнение массива:

1. Как постоянная:

```
Const  
b: array[1..10] of integer = (2, 4, 6, 8, 10, 12,14, 16, 18, 20);  
g: array[1..5] of char = ( 'a', 'b', 'c', 'd', 'e');  
Var  
a: array[1..5] of integer;  
Begin  
a[1]:=6; a[2]:=23; a[3]:=14; a[4]:=56; a[5]:=34;  
.....  
.....  
End.
```

2. С клавиатуры.

```
Var  
Mas: array[1..100] of integer;  
i : integer;  
Begin  
Writeln('введите размер массива');  
Readln(n); {Количество элементов массива}  
For I:=1 to n do  
begin  
writeln('введите ',I , ' –й элемент массива');  
readln(n);  
end;  
end.
```

3. С использованием датчиков случайных чисел.

```
Var  
Mas: array[1..100] of integer;  
i : integer;  
begin  
randomize; {инициализация встроенного генератора случайных чисел}  
Writeln('введите размер массива');
```

```
Readln(n);  
For i:=1 to n do  
begin  
mas[i]:=-5+random(15); {ввод значений массива, значение массива изменяется  
от -5 до 10}  
write(mas[i], ' '); {вывод элементов массива}  
end;  
end.
```

Задание на лабораторную работу

1. Задан одномерный массив B(10), заполненный произвольным образом. Подсчитать количество элементов массива, больших заданного числа k.
2. Создайте массив A[1..8] с помощью генератора случайных чисел с элементами от -10 до 10 и выведите его на экран. Подсчитайте количество отрицательных элементов массива.
3. Найти сумму положительных элементов линейного массива целых чисел. Размерность массива - 10. Заполнение массива осуществить с клавиатуры.
4. Дана последовательность целых положительных чисел. Найти произведение четных чисел.
5. Дан массив из 100 целых чисел. Найти сумму элементов массива.
6. Дан массив из 20 целых чисел. Все элементы меньше 10 заменить на 0 и вывести полученный массив на экран.
7. Определить самую высокую температуру и самый теплый день в мае.

Лабораторная работа № 14.
ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ДВУМЕРНЫХ
МАССИВОВ.

Цель работы: освоить вопросы объявления двумерных массивов в Pascal, научиться выполнять действия над элементами массивов.

Матрица или двумерный массив – это прямоугольная таблица чисел (или других элементов одного типа). Каждый элемент матрицы имеет два индекса (номер строки и номер столбца).

Объявление двумерного массива в Pascal:

```
var A: array[1..3,1..4] of integer;
```

Варианты описания двумерного массива

1. Описание массива в разделе переменных:

```
const N = 3;  
      M = 4;  
var A: array[1..N,1..M] of integer;
```

2. Описание массива через раздел type:

```
const  
      M=10;  
      N=5;  
type  
matrix=array [1..M, 1..N] of integer;  
var A: matrix;
```

3. Ввод двумерного массива m x n с клавиатуры:

```
for i:=1 to N do  
  for j:=1 to M do begin  
    write('A[' ,i ,',' ,j ,']=');  
    read ( A[i,j] );  
  end;
```

4. Заполнение случайными числами:

```
for i:=1 to N do begin  
  for j:=1 to M do  
    write ( A[i,j]:5 );  
  writeln;  
end;
```

Задание на лабораторную работу

1. Найти сумму элементов массива (прямоугольной таблицы) размером $[4 \times 5]$
2. Найти максимальный элемент массива размером $[7 \times 7]$.
3. Сформировать матрицу размерностью 5×5 . Вывести ее на экран. Затем преобразовать ее, заменив элементы, у которых $n=m$ на 0.
4. Составить программу, позволяющую с помощью датчика случайных чисел сформировать матрицу размерностью $[10 \times 10]$. Определить: минимальный элемент, лежащий ниже побочной диагонали; произведение ненулевых элементов последней строки.
5. В двумерном массиве размером $[5 \times 5]$ определить среднее значение элементов. Найти индекс элемента массива, наиболее близкого к среднему значению.

Лабораторная работа № 15.

ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СТРУКТУР.

Цель работы: изучить основы программирования с использованием динамических структур данных в Pascal.

Объект данных обладает динамической структурой, если его размер изменяется в процессе выполнения программы или он потенциально бесконечен.

Стеки

Стек — динамическая структура данных, в которой добавление и удаление элементов доступно только с одного конца (с верхнего (последнего) элемента). Алгоритм работы со стеком характеризуется правилом: «последним пришел - первым вышел» (LIFO, last in - first out).

добавление в стек нового элемента;

`add(<нач_стека>,<новый_элемент>):<нач_стека>`

определение пуст ли стек;

`empty(<нач_стека>):boolean`

доступ к последнему включенному элементу, вершине стека;

`take(<нач_стека>):<тип_элементов_стека>`

исключение из стека последнего включенного элемента.

`del(<нач_стека>):<нач_стека>`

Пример. Создать процедуры для добавления элемента в стек (add), удаления элемента из стека (del), вывода содержимого стека на экран (wstack).

Решение

Описание переменных:

top — указатель на конец стека;

p — указатель на обслуживаемую в текущий момент область памяти.

Алгоритм решения задачи:

В теле основной ветки программы переменной top присваивается значение nil, т.е. стек еще не заполнен.

Добавление элемента в стек

Процедура `add` принимает в качестве фактического параметра число `x`, которое будет записано в содержательную часть элемента стека.

Выделяется память под указатель `p`, в содержательную часть которого записывается `x`, а в указатель – адрес, хранимый в `top`. Переменная `top`, в свою очередь, начинает ссылаться на выделенную под `p` область памяти.

Удаление элемента из стека

В процедуре `del` в переменную `p` записывается адрес предпоследнего элемента стека. Последний элемент, ссылка на который хранится в `top`, удаляется. Далее `top` присваивается адрес элемента, на участок памяти, который указан `p`. К этому времени он стал уже не предпоследним, а последним элементом.

Удаление не происходит, если стек пустой (`top` указывает на `nil`).

Вывод стека на экран

Сначала `p` указывает на последний элемент. Выражения цикла `while` будут выполняться до тех пор, пока в `p` не запишется `nil`; это будет означать то, что перед этим был обработан первый элемент стека (который обрабатывается последним). В цикле сначала на экран выводится содержательная часть динамической переменной, на которую указывает `p`. Затем значение `p` меняется: переменная начинает указывать на следующий элемент.

Задание на лабораторную работу

1. Набрать программу на языке Паскаль:

```
type
  pitem = ^item;
  item = record
    data: integer;
    prev: pitem
  end;

var
  top, p: pitem;
  n, k: integer;

procedure add(x:integer);
begin
  new(p);
  p^.data := x;
  p^.prev := top;
  top := p
end;
procedure del;
```

```

begin
  if top<>nil then begin
    p := top^.prev;
    dispose(top);
    top := p
  end;
end;
procedure wstack;
begin
  writeln('Содержимое стека: ');
  p := top;
  while p <> nil do begin
    write(p^.data, ' ');
    p := p^.prev;
  end;
  writeln;
end;

begin
  top := nil;
  for k := 1 to 10 do
    add(k);
  wstack;
  writeln('Введите добавляемое в стек значение: ');
  readln(n);
  add(n);
  wstack;
  writeln('Количество элементов стека для удаления? ');
  readln(n);
  for k:=1 to n do
    del;
  wstack;
  readln
end.

```

2. Объяснить каждую строчку в программе.

Примечания:

Основная ветка программы может быть любой. В приведенной выше программе сначала стек заполняется числами от 1 до 10 (процедура add вызывается 10 раз), затем они выводятся на экран. Пользователь вводит число, оно добавляется в стек. В конце программы указывается количество удаляемых элементов, после чего в цикле удаляется именно это количество элементов стека.

Лабораторная работа № 16. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СТРОК.

Цель работы: выработать навыки работы с символьной информацией в языке программирования Pascal, научиться использовать строки символов и множества при решении задач.

Переменные типа String аналогичны массивам типа Char. Их отличием является то, что число символов (длина строки) может динамически меняться в интервале от единицы до заданного верхнего значения.

Тип String (строка) в Паскале широко используется для обработки текстов. Этот тип является стандартным и во многом похож на одномерный массив символов Array [0..N] of Char. Значение N соответствует количеству символов в строке и может меняться от 0 до 255. Символы, входящие в строку, занимают позиции с 1 до N. Начальный байт строки с индексом 0 содержит информацию о ее длине, т.е. это символ с кодом, равным длине строки.

Можно, также описывать переменные типа String[K], где K - целое число не больше 255. Так определяются строки с длиной не больше K. Этот тип уже не является стандартным. С символами строки можно работать как с элементами массива из символов, но в отличие от массивов, строки можно вводить целиком, сравнивать друг с другом и сцеплять операцией "+".

Сравнение строк выполняется посимвольно в соответствии с их кодами до первого несовпадения. Если одна из строк закончилась до первого несовпадения, то она считается меньшей. Пустая строка меньше любой строки. ПРИМЕР: Сравнение строк.

```
'abcd' > 'abcD' { 'd' > 'D' }  
'abcd' > 'abc' { 'd' > "" }  
'abc' < 'axxc' { 'b' < 'x' }  
'abcd' = 'abcd'
```

Переменная строкового типа (String) может рассматриваться как массив элементов символьного типа (Char). Например, если в программе определены переменные S: string; C: char; и задано S:='Москва', то S[1]='М', S[2]='о' и т. д. и возможно присвоение, например: C:= S[1]; Таким образом строка может рассматриваться как линейный массив символов. Элементы массива, составляющие строку можно переставлять местами и получать новые слова, например:

```
for i:= 1 to N div 2 do  
begin  
  C:= S[i];  
  S[i]:= S[N-i+1];  
  S[N-i+1]:= C  
  Writeln(S);  
end; { исходное слово выведется справа налево: "авксом" }
```

Здесь $N := \text{ord}(S[0])$; - число символов в переменной "S" хранится в переменной S[0]. Функция "ord" преобразует символьный тип в целый. $N \div 2$ - количество перестановок для слова из "N" символов. В переменной "C" запоминается значение i-го элемента, который меняется с элементом, симметричным относительно середины строки.

Можно производить поиск и замену заданного символа в строке, например:

```
for i:=1 to N do if S[i]=' ' then writeln ('найден символ пробел');  
for i:=1 to N do if S[i]='/' then S[i]:='\'; {замена символа "/" на "\"}
```

Заменяя или переставляя символы в строке по определенной схеме (закону) можно зашифровать строку. Для дешифровки используется, как правило, схема обратной перестановки или замены символов. Например:

```
for i:=1 to N do S[i]:=chr(ord(S[i])+2);
```

{преобразование исходных символов в символы с кодом большим на две единицы}

Напомним, что все используемые в MS-DOS символы имеют ASCII коды от 0 до 255. Здесь удобно также использовать функции Pred(C); и Succ(C);

Существует ряд стандартных функций и процедур для работы со строками.

- Функция Length(s) выдает длину строки s.
- Функция Concat(s1,s2,...,sn) возвращает строку s1+s2+...+sn.
- Функция Copy(s,p,k) возвращает фрагмент строки s, который начинается в позиции p и имеет длину k.
- Функция Pos(s1,s) ищет первое вхождение подстроки s1 в строку s и возвращает номер первого символа s1 в строке s или 0 если не нашли.
- Процедура Delete(s,p,k) удаляет из строки s фрагмент, который начинается в позиции p и имеет длину k.
- Процедура Insert(s,s1,p) вставляет в строку s1 подстроку s, начиная с заданной позиции p.

Задание на лабораторную работу

1. Вывести строку длины N (N - четное), которая состоит из чередующихся символов C1 и C2, начиная с C1.
2. Дана строка. Вывести строку, содержащую те же символы, но расположенные в обратном порядке.
3. Дана строка. Если она представляет собой запись целого числа, то вывести 1; если вещественного (с дробной частью), то вывести 2; если строку нельзя преобразовать в число, то вывести 0.
4. Дана строка S и число N. Преобразовать строку S в строку длины N следующим образом: если длина строки S больше N, то отбросить первые символы, если длина строки S меньше N, то в ее начало добавить символы "." (точка).

5. Даны два числа: $N1$ и $N2$, и две строки: $S1$ и $S2$. Получить из этих строк новую строку, объединив $N1$ первых символов строки $S1$ и $N2$ последних символов строки $S2$.
6. Даны две строки: $S1$ и $S2$. Определить количество вхождений строки $S2$ в строку $S1$.
7. Даны строки $S1$, $S2$ и символ C . После каждого вхождения символа C в строку $S1$ вставить строку $S2$.
8. Даны две строки: $S1$ и $S2$. Удалить из строки $S1$ все подстроки, совпадающие с $S2$. Если таких подстрок нет, то вывести $S1$ без изменений.
9. Даны три строки: $S1$, $S2$, $S3$. Заменить в строке $S1$ первое1|последнее2|все3 вхождения строки $S2$ на $S3$.
10. Дана строка, состоящая из русских слов, разделенных пробелами (одним или несколькими). Определить количество слов в строке.

Лабораторная работа № 17. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ.

Цель работы: выработать навыки работы с символьной информацией в языке программирования Pascal, научиться использовать строки символов и множества при решении задач.

Функция в Паскале — это подпрограмма, которая в отличие от процедуры всегда возвращает какое-либо значение. Для этого в теле функции её имени присваивается вычисленное значение — результат, который она возвращает.

Особенности функций:

1. Функция имеет только один результат выполнения. Этот результат обозначается именем функции и передается в вызвавшую эту функцию программную единицу.
2. Для функции обязательно указывается ее тип. Тип может быть простым (скалярным) или строковым.

Описание функции в Паскале имеет вид:

```
Function Имя функции (формальные параметры):Тип результата;  
Раздел описаний  
Begin  
.....  
End;
```

Вызов функции производится по ее имени с указанием фактических параметров. Аналогично процедурам, фактические и формальные параметры в функции могут отсутствовать.

Рассмотрим пример создания программ с использованием функций. Пусть необходимо произвести следующие вычисления:

$$S = \frac{\sum_{i=1}^n x_i + \sum_{j=1}^m y_j}{a^n + b^m},$$

где где a , b , n , m - целые переменные;

$\{x_i\}$, $\{y_j\}$ - массивы, содержащие n и m вещественных чисел, соответственно.

В Паскале нет стандартных функций суммирования элементов массива и возведения чисел в степень больше 2. Разработаем свои функции для решения этих задач и будем использовать их для решения поставленной задачи.

```
Текст программы  
Uses crt;
```

```

Type
  Tmas=array[1..100] of real;
Var a,b,n,m,i,j:byte;
  X,Y:Tmas;
  S:Real;
Function Summa(Dlmas:byte; Mas:Tmas):Real;
  Var Sum:real; i:byte;
  Begin
    Sum:=0.0;
    For i:=1 to Dlmas do
      Sum:=Sum+Mas[i];
    Summa:=Sum;
  End;
Function step(Pok:byte;Osn:byte):real;
  Var i:byte; St:real;
  Begin
    St:=Osn;
    For i:=2 to Pok do
      St:=St*Osn;
    step:=St;
  End;
{ Главная программа }
BEGIN
  Write(' Введите длину первого массива - N ');
  Readln(N);
  Write(' Введите длину второго массива - M ');
  Readln(M);
  Writeln(' Введите элементы массива X');
  For i:=1 to N do Read(X[i]); readln;
  Writeln(' Введите элементы массива Y');
  For j:=1 to M do Read(Y[j]); readln;
  Write(' Введите a и b '); Readln(a,b);
  S:=(Summa(N,X)+Summs(M,Y))/(step(N,a)+step(M,b));
  Writeln(' Полученный результат S = ',S);readkey;
END.

```

Задание на лабораторную работу

1. Составить функцию, которая определяет сумму всех чисел от 1 до N и привести пример ее использования.
2. Составить функцию, которая определяет, сколько зерен попросил положить на N-ую клетку изобретатель шахмат (на 1-ую – 1 зерно, на 2-ую – 2 зерна (в два раза больше предыдущего результата), на 3-ю – 4 зерна, на 4-ю – 8 зерен...)
3. Составить функцию вычисления степени числа на языке Паскаль.

Лабораторная работа № 18.
ПОИСК ИНФОРМАЦИИ В ГЛОБАЛЬНОЙ СЕТИ: КАТАЛОГАХ,
ЭЛЕКТРОННЫХ БИБЛИОТЕКАХ И СПРАВОЧНИКАХ.

Цель работы: научиться искать учебную литературу в электронных библиотечных системах.

В Муромском институте ВлГУ организован доступ к следующим электронно-библиотечным системам (ЭБС):

Электронная библиотечная система «Айбукс»

ibooks.ru - авторизированный доступ из любой точки с доступом в Интернет (регистрация студентов и преподавателей осуществляется с компьютеров вычислительной сети вуза) «Айбукс» - это широкий спектр самой современной учебной и научной литературы ведущих издательств России. ЭБС постоянно пополняется электронными версиями изданий, только что вышедших из печати. Большинство книг имеют грифы Минобрнауки РФ, Учебно-методических объединений и Научно-методических советов по различным областям знаний.

Электронная библиотечная система «IPRBooks»

iprbookshop.ru

Электронно-библиотечная система IPRbooks — научно-образовательный ресурс для решения задач обучения в России и за рубежом. Уникальная платформа ЭБС IPRbooks объединяет новейшие информационные технологии и учебную лицензионную литературу. Контент ЭБС IPRbooks отвечает требованиям стандартов высшей школы, СПО, дополнительного и дистанционного образования. ЭБС IPRbooks в полном объеме соответствует требованиям законодательства РФ в сфере образования.

ЭБС IPRbooks обеспечивает возможность работы с постоянно пополняемой базой лицензионных изданий по широкому спектру дисциплин — учебные, научные издания и периодика, представленная более 300 федеральными, региональными и вузовскими издательствами, научно-исследовательскими институтами и ведущими авторскими коллективами.

ЭБС IPRbooks предназначена для студентов, аспирантов, преподавателей, научных работников и практикующих специалистов, стремящихся получать знания из качественных лицензионных источников.

Авторизированный доступ из любой точки с доступом в Интернет (регистрация студентов и преподавателей осуществляется с компьютеров вычислительной сети вуза).

Для работы с "Платформой библиокомплектатор" необходимо перейти по ссылке. При работе за пределами института необходимо пройти авторизацию.

Электронная библиотечная система «BOOK.RU»

BOOK.RU - лицензионная библиотека, которая содержит более 6000 наименований учебных и научных изданий от преподавателей ведущих вузов России. Фонд электронной библиотеки комплектуется на основании новых ФГОС ВО, СПО.

Библиотека регулярно пополняется новыми изданиями. На сайте размещаются книги до выхода их печатных аналогов.

Чтение электронной версии книг доступно в постраничном режиме, возможно цитирование до 10% содержания книги, а также создание конспекта на основе нескольких изданий.

Авторизированный доступ из любой точки с доступом в Интернет (регистрация студентов и преподавателей осуществляется с компьютеров вычислительной сети вуза).

Задание на лабораторную работу

1. Зарегистрироваться в электронных библиотечных системах.
2. Найти в каждой библиотечной системе учебники по информатике их библиографического списка.

Лабораторная работа № 19. ОСНОВЫ ЯЗЫКА HTML.

Цель работы: познакомиться со структурой html-документа; изучить основные тэги форматирования текста.

HTML (Hyper Text Markup Language) – язык разметки гипертекста. Для разметки HTML документа используют тэги (флаги разметки).

Тег – это определенная последовательность символов, заключенные между знаками < (больше) и > (меньше).

Для того, чтобы создать HTML документ необходимо:

- открыть любой текстовый редактор (например, блокнот встроенный в Windows);
- набрать произвольный текст и разметить его HTML тэгами;
- сохранить файл с расширением .htm или .html.

Теперь если открыть созданный файл с помощью веб-браузера, он будет отображен как веб-страница.

Структура HTML-документа:

```
<HTML>  
<HEAD>  
<TITLE>простое название</TITLE>  
</HEAD>  
<BODY>  
Содержание страницы  
</BODY>  
</HTML>
```

Параметры документа задаются через тег <BODY>. Параметры – это специальные символьные команды, которые «понимает» браузер. Тег BODY имеет следующие атрибуты:

- bgcolor – цвет фона задан по схеме RGB;
- background – фон – графический файл;
- text – цвет текста;
- link – цвет ссылки;
- alink – цвет активной ссылки;
- vlink – цвет посещенной ссылки.

Шрифт задается тегом FONT, который имеет следующие атрибуты: face – имена шрифтов, разделенные запятыми, size – размер от 1 до 7 (по умолчанию 3), color – цвет шрифта.

В HTML существуют специальные теги для заголовков: от H1 (самого крупного) до H6 (самого мелкого).

Списки на Web-страницах бывают маркированные (– начало текста списка, – начало каждого элемента в списке) и нумерованные (– начало текста списка, – начало каждого элемента в списке). Маркером списка может быть рисунок.

Задание на лабораторную работу

1. Создайте личную папку, куда вы будете сохранять все файлы своего сайта.
2. Запустите программу Блокнот (Notepad).
3. Наберите в окне программы простейший файл HTML.

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
Расписание занятий на вторник
</BODY>
</HTML>
```

4. Сохраните файл под именем RASP.HTML (обязательно укажите тип файла HTML при сохранении) в личной папке.

5. Для просмотра Web-страницы используйте любую программу браузера (Internet Explorer, Opera, Mozilla Firefox или другую). Для этого, не покидая программу Блокнот (сверните окно на панель задач), откройте личную папку и двойным кликом по файлу RASP.HTML откройте окно браузера.

6. При необходимости откройте текст Web-страницы в Блокноте (1 щелчок правой клавишей мыши по файлу RASP.HTML, в контекстном меню выбрать команду Открыть с помощью... и выбрать программу Блокнот). При необходимости открыть файл в браузере – двойной клик по значку файла левой клавишей мыши.

7. Внести изменения в файл RASP.HTML, расположив слова *Расписание, занятий, на вторник* на разных строках.

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
Расписание
занятий
на вторник
</BODY>
</HTML>
```

8. Сохраните текст с внесенными изменениями в файле RASP.HTML (меню Файл | Сохранить). Если у вас уже отображается Web-страница, то вам

достаточно переключиться на панели задач на программу браузера и обновить эту страницу (кнопка). Изменилось ли отображение текста на экране? Не удивляйтесь тому, что внешний вид вашей Web-страницы не изменился.

Не забывайте каждый раз сохранять текст Web-страницы при ее корректировке в программе Блокнот и обновлять страницу при ее просмотре в программе браузера.

Некоторые специальные команды форматирования текста

Существуют специальные команды, выполняющие перевод строки и задающие начало нового абзаца. Кроме того существует команда, запрещающая программе браузера изменять каким-либо образом форматирование текста и позволяет точно воспроизвести на экране заданный фрагмент текстового файла. Тег перевода строки
 отделяет строку от последующего текста или графики. Тег абзаца <P> тоже отделяет строку, но еще добавляет пустую строку, которая зрительно выделяет абзац.

Оба тега являются одноэлементными, тег <P> – двойной, т.е. требуется закрывающий тег.

9. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
<P>Расписание</P>
<BR>занятий<BR>
на вторник
</BODY>
</HTML>
```

10. Сохраните внесенные изменения, переключитесь на панели задач на программу браузера, обновите Web-страницу.

Выделение фрагментов текста

11. Внести изменения в текст файла RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
<B>Расписание</B>
<I> занятий</I>
<U> на вторник</U>
</BODY>
</HTML>
```

12. Посмотрите полученную Web-страницу.

Возможно использование комбинированных выделений текста.

`<I>Расписание</I>` `<I><U> занятий</U></I>` `<U> на вторник</U>`

Но при этом необходимо помнить следующее правило использования комбинированных тегов:

`<Тег_1><Тег_2> ... </Тег_2></Тег_1>` – правильная запись.

`<Тег_1><Тег_2> ... </Тег_1></Тег_2>` – ошибочная запись.

Обратите внимание на «вложенность» тегов, она напоминает «вложенность» скобок.

Установка размера текущего шрифта

Тег шрифта `` позволяет задавать размер текущего шрифта в отдельных местах текста в

диапазоне от 1 до 7.

13. Внесите изменения в текст RASP.HTML

`<HTML>`

`<HEAD>`

`<TITLE>Учебный файл HTML</TITLE>`

`</HEAD>`

`<BODY>`

`Расписание`

занятий на вторник

`</BODY>`

`</HTML>`

14. Самостоятельно измените размер текста «занятий на вторник», используя тег ``.

15. Измените оформление текста HTML-документа, используя тег выделения фрагментов и тег перевода строки и абзаца.

Лабораторная работа № 20.

СОЗДАНИЕ WEB-САЙТА НА ЯЗЫКЕ HTML.

Цель работы: создать свою страницу на основе html-документа; изучить основные тэги форматирования текста и использовать их при создании страницы.

Как подготовить хорошую Web-страницу

1. Следует обратить внимание на простоту и логичность расположения информации на ваших страницах. Один из способов сделать информацию более легкой для восприятия – оставить на странице достаточно свободного места, не содержащего ни текста, ни рисунков. Страница, содержащая много информации, только отпугнет посетителя.
2. Постарайтесь представить информацию в виде списков или таблиц так, чтобы можно было достаточно легко найти важные сведения.
3. Не размещайте одно изображение сразу за другим. Попробуйте распределить их по документу, оставив достаточно свободного пространства.
4. Информация должна размещаться частями, легкими для восприятия. Обратите внимание на длину абзацев. Если абзац слишком длинный, разбейте его на несколько небольших абзацев.
5. Если Web-страница имеет большой объем, то, возможно, вам следует вставить ссылки, позволяющие пользователю быстро перемещаться между частями одного документа. Иногда имеет смысл вместо одного длинного документа подготовить одну страницу, содержащую перечень тем, каждую из которых раскрыть на отдельной Web-странице, и установить ссылки на соответствующие Web-страницы.
6. Использование графики может дополнительно привлечь пользователей. Но необходимо помнить о времени загрузки вашей страницы, которое определяется количеством и объемом графической информации. Красивая картинка не произведет никакого впечатления, если для того, чтобы ее увидеть, придется долго ждать, пока она загрузится.

Тестирование

Перед тем как выставлять свои Web-страницы на сервер необходимо их протестировать. Созданные документы должны пройти «локальную проверку» в пределах вашего жесткого диска. При проверке используйте разные браузеры. Вы увидите различия, которые могут оказаться довольно существенными.

В рамках тестирования необходимо сделать следующее:

1. Проверить правописание. Выполните автоматизированную проверку правописания текста (для этого можно использовать Microsoft Word) или попросите кого-нибудь выполнить корректуру.
2. Проверить навигацию. Убедитесь, что на каждой странице присутствуют необходимые средства навигации, все ссылки работают правильно.

3. Проверить доступ к внешним файлам. Выясните, размещены ли графические, звуковые или видеофайлы там, где они могут быть найдены и откуда их можно загрузить (должен быть правильно указан путь доступа). Для неграфических браузеров требуется задать подменяющие текстовые сообщения.
4. Проверить, допустимо ли время загрузки.
5. Осуществить проверку ваших Web-страниц посторонним лицом. Попросите кого-нибудь, кто не знаком с вашими документами, пройти их от начала до конца. Иногда при этом выясняются такие факты, каких вы сами ни за что бы не заметили.

Задание на лабораторную работу

Создать html-страницу, содержащую информацию о Вас (ФИО, адрес, краткая биография, хобби, любимые книги, фильмы и др.). Обязательно применить оформление текста. Приветствуется использование фотографий.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Основы информатики: учебник / В.Ф. Ляхович, В.А. Молодцов, Н.Б. Рыжикова. – Москва : КноРус, 2016. – 347 с. – (Среднее профессиональное образование). – ISBN 978-5-406-04695-1.
2. Информатика: учебник / Угринович Н.Д. – Москва: КноРус, 2018. – 378 с. – (Среднее профессиональное образование).
3. Информатика. Практикум: практикум / Н.Д. Угринович. — Москва : КноРус, 2018. — 264 с. — Для СПО. — ISBN 978-5-406-06186-2.
4. Информационные технологии. Задачник: учебное пособие / С.В. Синаторов. – Москва: КноРус, 2017. – 253 с. – (Среднее профессиональное образование). – ISBN 978-5-406-04886-3. <https://www.book.ru/book/920544>
5. Базы данных: учебник / И.А. Кумскова. – Москва : КноРус, 2016. – 399 с. – (Среднее профессиональное образование). – ISBN 978-5-406-04521-3. <https://www.book.ru/book/919609>