ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

**КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ**

**ТЕХНОЛОГИИ**

***Методические рекомендации к лабораторным работам студентов специальности 1-53 01 02 “Автоматизированные системы обработки информации” дневной и заочной формы обучения***

**Часть 1**

Могилев 2018

УДК

ББК

И

Рекомендовано к изданию

учебно-методическим отделом

Белорусско- Российского университета

Одобрено кафедрой «Автоматизированные системы управления» « 25 » мая 2018г., протокол № 13

Составители: ст. преподаватель Т. Л. Шебан

Рецензент ст. преподаватель

Изложены рекомендации к выполнению лабораторных работ по дисциплине «Компьютерные информационные технологии».

Учебно-методическое издание

**КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

Ответственный за выпуск А.И. Якимов

Технический редактор

Компьютерная верстка

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.

Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 115 экз. Заказ №

Издатель и полиграфическое исполнение:

Государственное учреждение высшего профессионального образования

«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,

изготовителя, распространителя печатных изданий

№ 1/156 от 24.01.2014.

Пр.Мира, 43, 212000, Могилев.

© ГУ ВПО «Белорусско-Российский

университет», 2018

**Содержание**

Лабораторная работа 1 Форматирование документа с использованием стилей4

Лабораторная работа 2 Форматирование документа со сложной структурой 6

Лабораторная работа 3 Разработка таблиц в MS ACCESS **9**

Лабораторная работа 4 Создание запросов в MS ACCESS 13

Лабораторная работа 5 Разработка форм и отчетов в MS Access 18

Лабораторная работа 6 Базы данных в MS Excel. Фильтры 20

Лабораторная работа 7 Консолидация данных. Сводные таблицы и сводные диаграммы23

Лабораторная работа 8 Ввод-вывод данных. Разработка программы линейной структуры 24

Лабораторная работа 9 Разработка программы разветвленной структуры 28

Лабораторная работа 10 Оператор цикла с параметром For…Next. Обработка массивов 31

Лабораторная работа 11 Разработка программ циклической структуры 34

Лабораторная работа 12 Обработка строковых данных 37

Лабораторная работа 13 Разработка программ с использованием подпрограмм и функций40

Лабораторная работа 14 Разработка пользовательских форм43

Лабораторная работа 15 Разработка программ для работы с текстовыми данными 44

Список литературы48

Лабораторная работа 1 Форматирование документа с использованием стилей

*Цель работы*: научиться использовать библиотеку стилей Microsoft Word, создавать пользовательские стили, создавать оглавление, форматировать документ с использованием стилей на примере оформления пояснительной записки к курсовому проекту.

**Стили** представляют собой наборы команд форматирования. При создании стиля пользователь указывает значения отдельных параметров форматирования, которые должны быть включены в создаваемый стиль, для последующего применения всех этих параметров совместно посредством выбора имени этого стиля.

**Стили** определяют форматирование символов, текстовых фрагментов, абзацев, строк таблиц или уровней структуры документа. Существует два различных типа стилей:

**Стиль символа** - содержит параметры форматирования символов, включая шрифт, размер, начертание, положение и интервалы

**Стиль абзаца** - содержит параметры форматирования абзацев, такие как междустрочные интервалы, отступы, выравнивание и позиции табуляции

Экспресс-стили Word находятся на панели "Стили" лента "Главная"

**Определение (переопределение) стиля**

* Придайте фрагменту текста необходимый вид
* Откройте экспресс-галерею стилей и щелкните кнопкой мыши "Сохранить выделенный фрагмент как новый экспресс-стиль" (при изменении стиля - "Обновить стиль в соответствии с выделенным фрагментом)
* В появившемся окне введите имя нового стиля и нажмите ОК

**Создание (изменение) параметров стиля**

* Откройте окно "Стили"
* Нажмите кнопку "Создать стиль" (две буквы А)
* В появившемся окне "Создание стиля" настройте все необходимые параметры форматирования (при изменении стиля, соответственно "Изменение стиля")

Для облегчения работы со стилями существует специальный механизм - **инспектор стилей**, который позволяет отслеживать используемые в документе стили абзаца и текста. Для вызова инспектора стилей служит кнопка с изображением буквы "А" и увеличительного стекла

**Задание**

Текст пояснительной записки находится в файле Записка.docx

1. Создать большой и малый текстовый штамп.

2. Использовать их в документе 2 способами: как стандартный блок в колонтитулах и как подложку.

3. Оформить пояснительную записку одним документом в соответствии с требованиями.

4. Нумерацию разделов и подразделов оформить многоуровневым нумерованным списком (в разделе 1 – 3 уровня)

5. Каждый заголовок раздела и подраздела должны быть оформлены стилем «Раздел» и «Подраздел» соответственно (стиль абзаца). Стили создать самостоятельно.

6. Текст пояснительной записки должен быть оформлен стилем «Текст записки» (стиль текста).

7. Просмотреть документ в режиме структуры документа.

8. Средствами Word составить оглавление, оформить ссылки на разделы и подразделы.

9. Последний лист –Приложение – пустой, ориентация альбомная.

Пояснительная записка включает следующие структурные элементы.

Титульный лист

Содержание

Введение

* + - 1. Постановка задачи

1.1 Математическая модель задачи

1.2 Входные данные

1.3 Выходные данные

1.4 Обработка ошибок

* + - 1. Проектирование программного модуля
  1. Структурная диаграмма программного модуля
  2. Разработка схемы программного модуля и ее описание
  3. Разработка пользовательского интерфейса
     + 1. Реализация программного модуля
  4. Код программы
  5. Описание используемых операторов и функций
     + 1. Тестирование программного модуля

Заключение

Список использованных источников

Приложения

Поле <Шифр> титульного листа имеет следующую структуру:

301.<шифр специальности>.<№ зачетки>.<№ по списку>.81-01, где 301 – код вуза; <шифр специальности> - определен для каждой специальности, например, 1-53 01 02 – «Автоматизированные системы обработки информации».

Пояснительная записка выполняется на листах формата А4 с рамкой и текстовым штампом (один большой текстовый штамп (рисунок 1.1) под содержание, остальные – малые (рисунок 1.2)).

Вся пояснительная записка оформляется единым шрифтом (Times New Roman, 14 пт). Междустрочный интервал – полуторный, отступ первой строки –1,25 см, выравнивание – по ширине, расстояние между текстом и рамкой 0.5 см. Каждый раздел пояснительной записки начинается с нового листа. Разделы и подразделы ведутся с абзаца.

Разделы и подразделы 1-го и 2-го уровня (1, 1.1 ) отделяются вертикальными отступами (12 пт), для 3-го уровня и ниже (1.1.2) – обычный межстрочный интервал

Содержание, введение, заключение и список использованных источников не нумеруются (и, как правило, центрируются)

Нумерация страниц сквозная и ведется в штампе. Титульный лист учитывается как первый, но номер страницы на нем не проставляется. Задание в нумерации страниц не учитывается.

Нумерация рисунков сквозная. Подпись рисунка отделяется вертикальными отступами 12 пт от рисунка и следующего за ним текста. Для рисунков, расположенных на нескольких листах, ведется нумерация листов, например, Рисунок 4. Лист1 – Название.

У

*2*

*24*

Листов

*Изм.*

Лист

№ докум.

Подп.

Дата

Разраб.

Пров.

*Н.контр.*

*Утв.*

Лит.

Лист

301 гр. АСОИ-151

*Иванов*

Петров

<Шифр>

###### Разработка программного модуля

Рисунок 1.1 – Большой текстовый штамп

Рисунок 1.2 – Малый текстовый штамп

Изм.

Лист

№ докум.

Подп.

Дата

<Шифр>

Лист

Лабораторная работа 2 Форматирование документа со сложной структурой

*Цель работы*: Освоение основных приемов работы с графическими примитивами, надписями, формулами, расположение их в тексте документа.

Инструменты для работы с графикой находятся на панели **"Иллюстрации"** ленты **"Вставка"**.

***Создание графического примитива***

Кнопка *"Фигуры"* служит для быстрого создания графических примитивов. Для создания нужного примитива надо его выбрать из выпадающего списка и "нарисовать" в документе протяжкой мыши с нажатой левой кнопкой. Для того, чтобы фигура имела правильные пропорции, во время рисования надо удерживать нажатой кнопку Shift.

Когда фигура нарисована, появляется контекстный инструмент **"Средства рисования"** с лентой **"Формат"**.

Окно панели **"Стили фигур"** содержит расширенные параметры форматирования *"Формат автофигуры"*. В этом окне можно произвести большинство настроек форматирования.

Наиболее часто встречающиеся настройки вынесены на ленту **"Формат"**.

Инструменты, расположенные на панели **"Упорядочить"** предназначены для настройки параметров взаимодействия фигуры с текстом документа.

Кнопка *"Положение"* задает расположение графического объекта на странице. Для настройки обтекания фигуры текстом служит кнопка *"Обтекание текстом"*.

Если в документ вставлено несколько фигур, перекрывающих друг друга, то их относительный порядок размещения можно настроить при помощи кнопок *"На передний план"* и *"На задний план"*.

***Группировка фигур***

Случаются такие ситуации, когда в документе размещены несколько объектов и с ними одновременно нужно произвести какие-либо действия (увеличить, уменьшить, переместить). В этом случае целесообразно произвести группировку объектов.

Для группировки фигур их необходимо предварительно выделить. Это можно осуществить при помощи кнопки *"Выделить"* на ленте **"Главная"**.

После этого надо перейти на панель **"Упорядочить"** и воспользоваться кнопкой *"Группировать"*.

Все выделенные объекты становятся, как бы, одним объектом, о чем свидетельствуют угловые маркеры. Теперь можно производить с ними все необходимые действия. После этого (при необходимости) объекты можно разгруппировать.

***Работа с надписями***

Особым видом графического примитива является *Надпись*. Этот примитив может содержать "в себе" текст.

Такие графические элементы, содержащие текст, можно связывать между собой. Для этого их необходимо предварительно разместить в документе. Затем выделить надпись, с которой будет начинаться текст. После этого на панели **"Текст"** воспользоваться кнопкой *"Создать связь"*. Курсор примет вид кружки. Подвести курсор к надписи, следующей за главной (при этом кружка начнет "выливаться") и нажать левую кнопку мыши. Теперь текст будет перетекать из одной надписи в другую.

***Объекты SmartArt***

Графика SmartArt позволяет быстро создавать разнообразные красочные схемы.

При выборе шаблонов SmartArt необходимо учитывать их первоначальное предназначение. Для вставки объекта SmartArt служит одноименная кнопка на панели **"Иллюстрации"** ленты **"Вставка"**. Откроется окно *"Выбор рисунка"* .Выбрав шаблон, вы увидите краткое его описание.

После добавления шаблона в документ в окне текстового процессора появится контекстный инструмент **"Работа с рисунками SmartArt"**, содержащий две ленты: **"Конструктор"** и **"Формат"**.

Для удаления какого-либо элемента необходимо его выделить и нажать клавишу Delete. Кнопки *"Повысить уровень"* и *"Понизить уровень"* предназначены для изменения уровня выделенных элементов.

Для форматирования объекта SmartArt предназначена лента **"Формат"** контекстного инструмента **"Работа с рисунками SmartArt"**.

***Коллекция Clip Art***

Данная коллекция содержит подборку набора картинок текстового редактора. Для вставки клипа необходимо нажать кнопку *"Клип"* на панели **"Иллюстрации"** ленты **"Вставка"**.У правого края окна появится панель **"Клип"**. Внизу находится кнопка *"Упорядочить клипы"*, нажав на которую мы попадем в окно *"Организатор клипов"*.Слева находится каталог клипов, а справа - область просмотра выбранного раздела каталога.

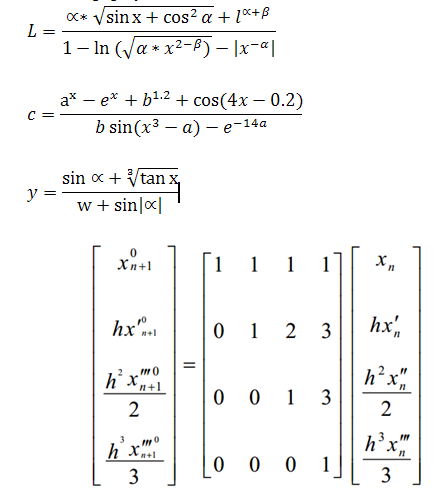
***Математические формулы.***

На вкладке **Вставка** в группе **Символы** щелкните стрелку рядом с пунктом **Формула**, а затем выберите подходящую формулу или выберите пункт **Вставить новую формулу**. При вставке новой формулы появляется контекстная лента **Работа с формулами и Конструктор**. **Конструктор** содержит все необходимые инструменты для создания и редактирования формул.

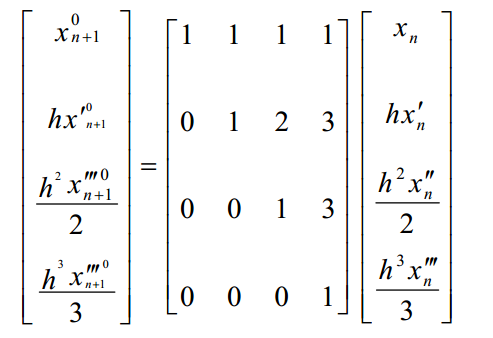
**Задание**

Создать новый документ, который должен содержать:

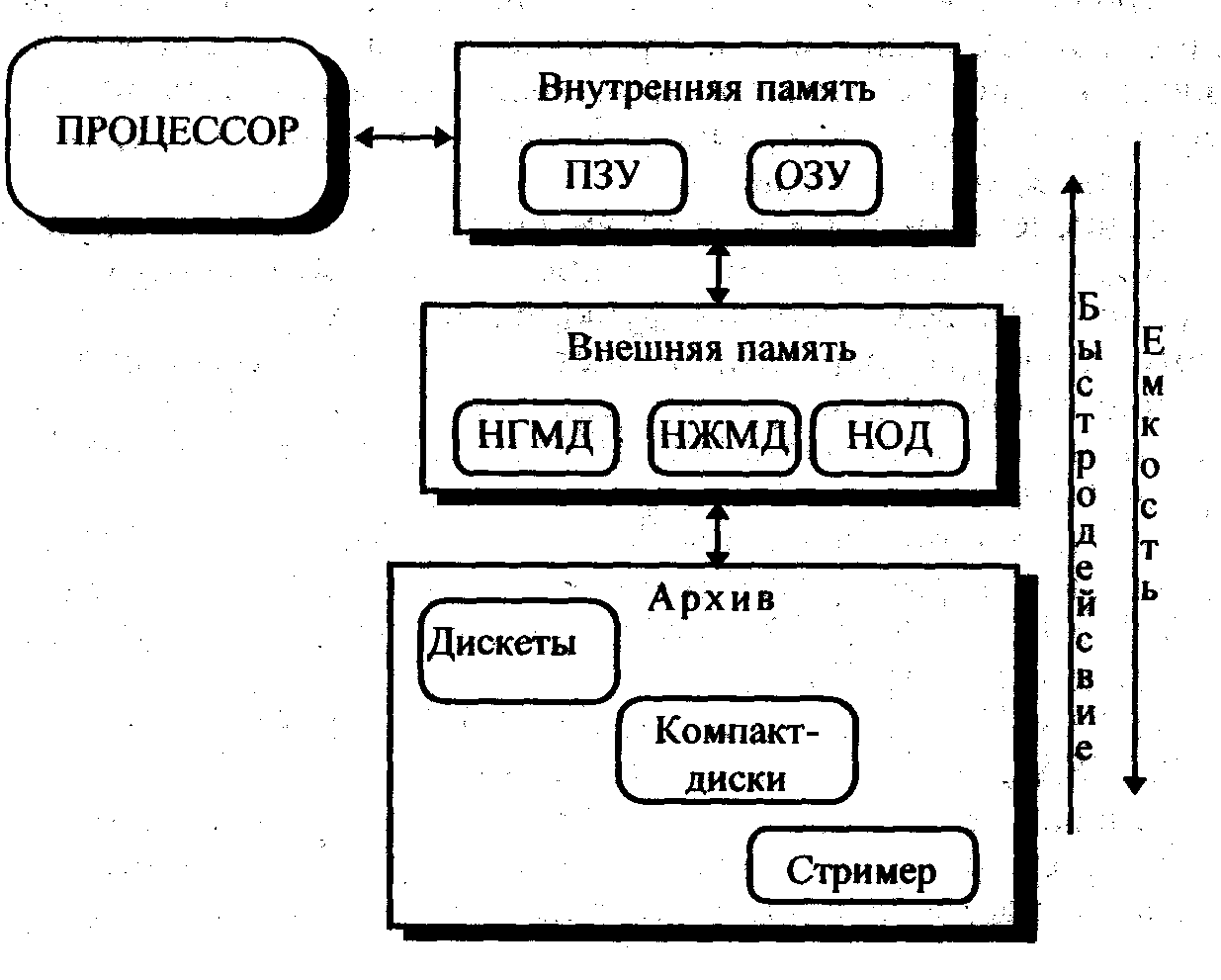
1. 3 формулы:



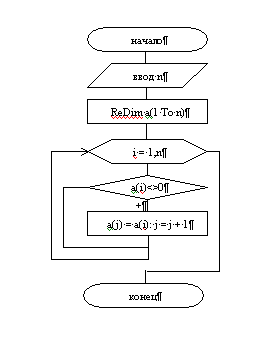
1. систему уравнений в матричной записи



1. Рисунок:



1. блок – схему:



1. Плакат «Преимущества работы в MS Office 2007» (использовать текст из документа *Преимущества работы в MS*Office 2007), содержащий:

* Графические примитивы
* Связанные надписи
* Объекты SmartArt
* Рисунки (сжать)
* Объекты WordArt

Лабораторная работа 3 Разработка таблиц в MS ACCESS

*Цель работы*: Освоение основных приемов работы с таблицами в среде MS ACCESS.

**База данных** – это реализованная с помощью компьютера модель данных, отражающая состояние объектов и их отношения. База данных (БД) предполагает наличие комплекса программных средств, обслуживающих ее и позволяющих использовать содержащуюся в ней информацию. Такие комплексы программ называют СУБД (системами управления базами данных).

**В Access база данных** – это файл, в котором хранятся все объекты, необходимые для обеспечения работы пользователя: таблицы, запросы, формы, отчеты, страницы доступа к данным, макросы и модули.

Все управление базой данных осуществляется в Окне базы данных*,* появляющемся при открытии БД. Это окно содержит основные объекты Access: таблицы, запросы, формы, отчеты, макросы и модули. В строке заголовка окна всегда отображается имя открытой или создаваемой БД.

Работать с таблицей MS ACCESS можно в **двух основных режимах**: в режиме конструктора и в режиме таблицы (можно выбрать из контекстного меню).

**Режим таблицы** используется для просмотра, добавления, изменения, сортировки и удаления данных.

В **режиме конструктора** задается структура таблицы, т.е. определяются типы, свойства полей, их число и названия (заголовки столбцов). Он используется для изменения только структуры таблицы. На рисунке 3.1 представлен вид таблицы Группы по маске1 базы данных Библиотека в режиме конструктора.

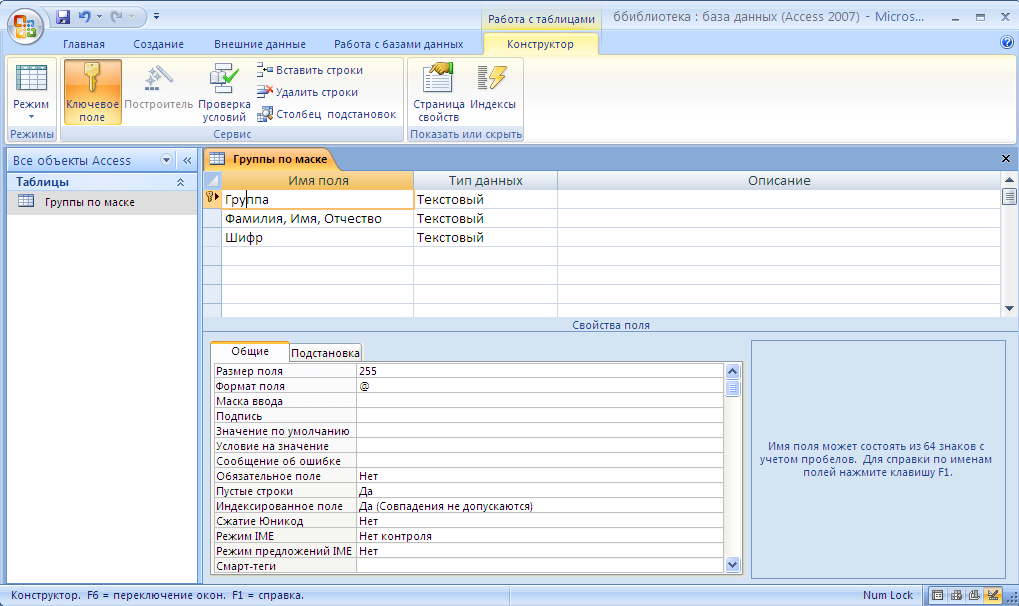


Рисунок 3.1 – Таблица в режиме конструктора

В режиме конструктора **верхняя часть окна** содержит список полей таблицы.

В столбце **Имя поля** указаны имена по­лей, выбранных при создании таблицы. Обратите внимание, что они не содержат пробелов. По имени будет осуществляться доступ к данным. Имя можно набирать русскими буквами.

В столбце **Тип данных** указан тип данных, содержащихся в поле.

В столбце **Описание** может быть представлено краткое описание каждого поля.

Обратите внимание на значок ключевого поля слева от поля Группа. **Ключевое поле** используется для однозначной идентификации каждой записи таблицы (в таблице не должно быть двух или более записей, имеющих одинаковое значение в ключевом поле). По ключевым полям таблицы можно связывать друг с другом.

В **нижней части окна** представлена таблица, содержащая свойства активного поля, в данном случае – поля Код книги. Рассмотрим основные из этих свойств.

**1*.*** **Размер поля**. Свойство Размер поля определяет размер или величину значений, ко­торые могут быть введены в это поле.

**2.** **Формат поля** (формат отображения) используется для вывода данных в формах и запросах. Существует набор специальных символов формата, которые задают вид и размер выводимых строк.

**3*.*** **Маска ввода** позволяет задать вид строки при наборе данных.

**4*.* Подпись поля**– это второе имя поля, которое можно использовать в таблицах и формах. Подпись может состоять из нескольких слов и быть более понятной, нежели обычное имя поля.

**5*.* Значение по умолчанию**. Чаше всего такой параметр необходим для числовых или логических полей, когда изначально необходимо наличие данных.

**6*.*** **Условие на значение** и **Сообщение об ошибке**. С помощью инструмента Построителя выражений можно сформировать ограничения на значения поля. Это должно быть выражение логического типа.

Если пользователь введет число, не удовлетворяющее этим условиям, про­грамма выведет на экран сообщение об ошибке. Для этого сообщения надо задать строку в параметре Сообщение об ошибке.

**7*.* Обязательное поле**. Если эта опция выбрана, при заполнении новой записи пользователь обязательно должен ввести какие-нибудь данные. В противном случае программа будет выводить «ругательные» сообщения, пока пользова­тель не введет данные.

**8*.* Индексированное поле.** Этот параметр может иметь одно из трех значений.

1. Поле не индексировано.

2. Поле индексировано, допускаются повторяющиеся значения.

3. Поле индексировано, не допускаются повторения значений поля.

Чаше всего это свойство используют для запрета повторяющихся значений

Существуют следующие способы создания пустой таблицы для последующего ее заполнения собственными данными.

* В мастере таблиц выбирают поля для новой таблицы из различных образцов таблиц, таких как деловые контакты, личное имущество и т. п.
* Создают таблицу в режиме конструктора, который позволяет добавлять поля, настроить отображение полей и обработку в них данных, а затем создают первичный ключ . Первичный ключ - одно или несколько полей (столбцов), комбинация значений которых однозначно определяет каждую запись в таблице. Первичный ключ не допускает значений Null и всегда должен иметь уникальный индекс. Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах..

Существуют следующие способы создания таблиц на основе существующих данных.

* Импорт или связывание данных из другой базы данных Microsoft Access или из файлов других программ.
* Создание таблицы на основании данных из текущей таблицы при помощи запроса на создание таблицы. Например, запросы на создание таблиц можно использовать для архивирования старых записей, резервного копирования таблиц, выбора групп записей для экспорта в другую базу данных или для использования в качестве основы для отчетов, отражающих данные в определенный момент времени.

Между таблицами могут быть установлены связи один-к-одному, один-ко-многим и многие-ко-многим. Информация о каждом типе связи имеется в справочной системе ACCESS. Поля таблиц, используемые для установления связей должны иметь одинаковый тип данных (допускается связь данных типа счетчик с числовым типом), их значения должны отвечать требованию целостности данных в базе данных.

Во всех лабораторных работах в среде Microsoft Access используется учебная база данных Борей.mdb.

**Задание**

1.Создание новой таблицы

* Создайте новую таблицу в режиме конструктора, включающую следующие поля: Код товара, Рекламируемый товар, Сотрудник, Дата начала компании.
* В качестве ключевого поля выберите Код товара. Для этого щелкните по кнопке “ Ключевое поля ” на панели инструментов. Сохраните таблицу под именем Реклама.
* Заполните таблицу Реклама 5-ю записями. При заполнении поля Сотрудник используйте коды реальных сотрудников фирмы Борей (уточните по таблице Сотрудники).

2. Установление связи между таблицами.

* Выберите вкладку Работа с базами данных - Схема Данных для создания связи между таблицами Сотрудники и Реклама. Добавьте в него таблицы Реклама и Сотрудники (или только Реклама , если Сотрудники уже там есть). Для этого нужно щелкнуть правой кнопкой мыши по свободному месту в окне “Схема Данных” и выбрать команду Добавить таблицу из всплывающего меню. В окне Добавление таблицы указать таблицу Реклама.
* Перетащите курсор мыши с нажатой левой клавишей от поля Код сотрудника таблицы Сотрудники на поле Сотрудник таблицы Реклама. Откроется окно “Изменение связей” и автоматически будет определен тип связи Один-ко-многим. Включить переключатель «Обеспечение целостности данных». Нажать кнопку “Создать”. Будет создана связь.

3. Удаление таблицы из базы данных.

* Удалите взаимосвязь между таблицами Сотрудники и Реклама (выделить связь и нажать клавишу Delete). Удалите таблицу Реклама из схемы данных.

4. Повторите создание таблицы Реклама1 в режиме конструктора, изменив тип поля Сотрудник. Вместо Числового типа **создайте столбец подстановки** фамилий сотрудников из таблицы Сотрудники, выбрав мастер подстановок в столбце Тип данных. Заполните таблицу 5 записями.

1. Поиск записей

Выполните поиск и просмотр тех записей, в которых фамилия начинается с “Кр”: В режиме Таблицы выделите поле Фамилия, а затем используйте кнопку “Поиск” (с изображением бинокля) на вкладке Главная группа Найти.

1. Использование фильтра для нахождения конкретных записей.

Укажите всех сотрудников фирмы, которые пришли на работу в фирму до сентября 1993 года. Используйте фильтр с указанием даты для поиска ответа на вкладке **Главная** группа **Сортировка и фильтр** из списка **Изменить параметры расширенного фильтра** выберите **Изменить фильтр**,

1. Сортировка записей с помощью кнопок на панели инструментов.

В режиме Таблицы щелкните в любой строке нужного столбца. Далее нужно отсортировать записи, щелкнув по кнопке По Возрастанию или По Убыванию.

Лабораторная работа 4 Создание запросов в MS ACCESS

*Цель работы*: Освоение основных приемов работы с запросами в среде MS ACCESS.

Запросы используются для просмотра, изменения и анализа данных различными способами. Запросы также можно использовать в качестве источников записей для форм, запросов, отчетов. В Microsoft Access есть несколько типов запросов.

***Запросы на выборку***

Запрос на выборку является наиболее часто используемым типом запроса. Запросы этого типа возвращают данные из одной или нескольких таблиц и отображают их в виде таблицы, записи в которой можно обновлять (с некоторыми ограничениями). Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений.

***Запросы с параметрами***

Запрос с параметрами — это запрос, при выполнении отображающий в собственном диалоговом окне приглашение ввести данные, например условие для возвращения записей или значение, которое требуется вставить в поле. Можно разработать запрос, выводящий приглашение на ввод нескольких единиц данных, например двух дат. Затем Microsoft Access может вернуть все записи, приходящиеся на интервал времени между этими датами.

Запросы с параметрами также удобно использовать в качестве основы для форм, отчетов и страниц доступа к данным. Например, на основе запроса с параметрами можно создать месячный отчет о доходах. При печати данного отчета Microsoft Access выводит на экран приглашение ввести месяц, доходы за который должны быть приведены в отчете. После ввода месяца Microsoft Access выполняет печать соответствующего отчета.

***Перекрестные запросы***

Перекрестные запросы используют для расчетов и представления данных в структуре, облегчающей их анализ. Перекрестный запрос подсчитывает сумму, среднее, число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк. Перекрестный запрос создается с помощью мастера или самостоятельно в режиме конструктора запроса. В бланке запроса нужно указать поля, значения которых будут заголовками столбцов и строк, а также поле, значения которого следует использовать в вычислениях. Существует возможность представления рассчитанных данных в денежном и других форматах.

***Запросы на изменение***

Запросом на изменение называют запрос, который за одну операцию изменяет или перемещает несколько записей.

Существует четыре типа запросов на изменение.

* *На удаление записи*.   Запрос на удаление удаляет группу записей из одной или нескольких таблиц. Например, запрос на удаление позволяет удалить записи о товарах, поставки которых прекращены или на которые нет заказов. С помощью запроса на удаление можно удалять только всю запись, а не отдельные поля внутри нее.
* *На обновление записи*.   Запрос на обновление вносит общие изменения в группу записей одной или нескольких таблиц. Например, на 10 процентов поднимаются цены на все молочные продукты или на 5 процентов увеличивается зарплата сотрудников определенной категории. Запрос на обновление записей позволяет изменять данные в существующих таблицах.
* *На добавление записей*.   Запрос на добавление добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких таблиц. Например, появилось несколько новых клиентов, а также база данных, содержащая сведения о них. Чтобы не вводить все данные вручную, их можно добавить в таблицу «Клиенты».
* *На создание таблицы*.   Запрос на создание таблицы создает новую таблицу на основе всех или части данных из одной или нескольких таблиц. Запрос на создание таблицы полезен при создании таблицы для экспорта в другие базы данных Microsoft Access или при создании архивной таблицы, содержащей старые записи.

***Запросы SQL***

Запрос SQL — это запрос, создаваемый при помощи инструкций SQL. Язык SQL (Structured Query Language) используется при создании запросов, а также для обновления и управления реляционными базами данных, такими как базы данных Microsoft Access.

Когда пользователь создает запрос в режиме конструктора запроса, Microsoft Access автоматически создает эквивалентную инструкцию SQL. Фактически, для большинства свойств запроса, доступных в окне свойств в режиме конструктора, имеются эквивалентные предложения или параметры языка SQL, доступные в режиме SQL. При необходимости пользователь имеет возможность просматривать и редактировать инструкции SQL в режиме SQL. После внесения изменений в запрос в режиме SQL его вид в режиме конструктора может измениться.

**Задание**

Используется учебная база данных Борей.

* 1. Создание запроса с помощью средства ACCESS “Простой запрос”.

Создайте запрос “Просмотр заказов” на базе таблицы «Заказы» в режиме Простой запрос.

* Перейдите на вкладку **Создание**, группа **Другие**, кнопка **Мастер запросов**, выберите режим **Простой запрос**.
* Выберите таблицу «Заказы».
* Переместите поля код заказа, код клиента, дата размещения, дата исполнения, название получателя из списка Доступные поля в список Выбранные поля.
* Сохраните запрос под именем “Просмотр заказов”.

2. Создание запроса на выборку в режиме конструктора.

Создайте запрос на базе таблицы Товары в режиме конструктора, содержащий сведения о кондитерских изделиях, цена которых не превышает 1500 руб. за единицу.

* Перейдите на вкладку **Создание**, группа **Другие**, кнопка **Конструктор** запросов, выберите таблицу «Товары».
* Включите в запрос поля код типа, марку, цену и поставщика товаров.
* Укажите условие отбора для кондитерских изделий в бланке запроса по типу под номером 3. Введите в бланк запроса выражение для отбора товара ценой за единицу не более 1500 руб.(<=1500)
* Выполните и сохраните запрос под именем “ Кондитерские изделия”.

3. Сортировка данных в запросе и поиск в заданном диапазоне.

Разместить в алфавитном порядке товары,. имеющиеся на складе в количестве не менее 20 единиц с ценами в интервале от 1000 руб. до 2000 руб.

* Создайте новый запрос на основе таблицы Товары.
* Включите в запрос поля, содержащие идентификатор товара, его марку, цену единицы и количество товара на складе.
* Выполните сортировку данных запроса по марке товара в алфавитном порядке.
* Используйте выражение с операторами сравнения для нахождения цен между 1000 и 2000 руб.(>=1000 and <=2000)
* Введите в запрос выражение, которое позволит отобрать лишь те товары, число которых на складе составляет не менее 20 единиц.
* Выполните и сохраните запрос под именем Товары умеренной стоимости.

4. Создание запроса на основе связанных таблиц.

Получите сведения о заказах из США и видах доставки.

* Создайте новый запрос, использовав таблицы Заказы и Доставка.
* Включите в поле Страна получателя условие отбора, которое позволило бы отобрать все заказы с доставкой в США.
* Включите в запрос поле с указанием названия доставки.
* Выполните и сохраните запрос под именем Доставка в США.

5. Суммирование данных в запросе.

Получить сведения о суммах, выплаченных за доставку заказов в США и Францию.

* Включите в новый запрос поля из таблицы Заказы, которые содержат сведения о стране получателя, доставке и затратах на нее.
* Выберите команду Итоги на панели инструментов, чтобы добавить в бланк запроса по образцу строку Групповые операции. После этого выделите в ячейке Группировка под полем Стоимость доставки строку SUM.
* Выполните и сохраните запрос

1. Включение в запрос параметров.

* Откройте в режиме конструктора запрос “Просмотр заказов”. Добавьте в ячейку Условие отбора под полем Дата исполнения заказа параметр, который в дальнейшем будет служить подсказкой при получении информации в диапазоне дат BETWEEN [ Начало периода ] AND [Конец периода]
* Выполните запрос и выведите на экран заказы, должны быть выполнены с 26 по 31 декабря 1996 года и сохраните его под именем “Заказы за период”.

1. Создание запроса с вычисляемыми полями.

Нужно рассчитать сумму товаров, хранящихся на складе.

* С помощью мастера построения простых запросов создайте запрос из таблиц Товары (поля Код товара, На складе, Цена) и Типы (поле Категория) и сохраните его под именем Вычисляемые поля.
* В режиме конструктора в строке Условие отбора поля Категория введите категорию «фрукты».
* В бланке запроса установите курсор в пустое поле. Нажмите кнопку «Построитель» на панели инструментов. Используя инструменты построителя выражений введите выражение для расчета суммы Сумма: [Цена]\*[НаСкладе].
* Выполните запрос и сохраните его.

1. Разработка запроса на создание таблицы

Требуется создать новую таблицу Неперспективные товары на базе таблицы Товары и включить в нее те товары, поставки которых прекращены.

* + В режиме конструктора создайте запрос на базе таблицы Товары, включив в него все поля этой таблицы
  + В поле ПоставкиПрекращены в строке Условие отбора укажите Да (так в запросе устанавливается значение Истина для логического типа данных, можно также использовать выражение Истина)
  + На вкладке **Конструктор**, группа **Тип** **запроса** выберите **Создание** **таблицы** и в появившемся диалоговом окне “Создание таблицы” укажите имя таблицы “Неперспективные товары”, выберите переключатель В текущей базе данных и нажмите кнопку ОК.
  + Выполните запрос. Появится сообщение о создании новой таблицы с указанием числа записей, помещаемых в нее. Нужно выбрать Да. В результате будет создана требуемая таблица, которую можно увидеть в группе Таблицы.

1. Создание запроса на обновление записей
   * В режиме конструктора создайте запрос на базе таблицы Товары, включив в него все поля этой таблицы
   * На вкладке **Конструктор**, группа **Тип** **запроса** выберите **Обновление**
   * В поле Цена в строке Обновление напишите выражение [Цена]+[Цена]\*0,1
   * В строке Условие отбора укажите для поля код типа 4, что соответствует Молочным продуктам
   * Выполните запрос и сохраните его.
   * Появится сообщение об обновлении записей с указанием их числа
2. Создание запроса на удаление записей.

Требуется удалить из таблицы Товары те товары, поставки которых прекращены. Это удобно сделать с помощью запроса на удаление записей**.**

* + В режиме конструктора создайте запрос на базе таблицы Товары, включив в него все поля этой таблицы В поле ПоставкиПрекращены в строке Условие отбора укажите Да (или Истина)
  + На вкладке **Конструктор**, группа **Тип** запроса выберите **Удаление**.
  + Выполните запрос и сохраните его.
  + Появится сообщение об удалении записей из таблицы с указанием числа удаляемых записей. Нужно выбрать Да.

1. Создание запроса на добавление записей

Добавим товары из таблицы Неперспективные товары, имеющиеся на складе, в таблицу Товары.

* + Создайте в режиме конструктора запрос, содержащий таблицу, записи из которой необходимо добавить в другую. Это будет запрос на базе таблицы Неперспективные товары.
  + В режиме конструктора запроса нажмите стрелку рядом с кнопкой **Тип запроса** на панели инструментов и выберите команду **Добавление**. На экране появится диалоговое окно **Добавление**.
  + В поле **Имя таблицы** введите имя таблицы, в которую необходимо добавить записи. Это будет таблица Товары

Выполните одно из следующих действий.

Если таблица находится в открытой в настоящий момент базе данных(это наш случай), выберите параметр **в текущей базе данных**.

* + Нажмите кнопку OK.
  + Перетащите из списка полей в бланк запроса поля, которые необходимо добавить или которые будут использоваться при определении условия отбора.

Если в обеих таблицах выделенные поля имеют одинаковые имена, то соответствующие имена автоматически вводятся в строку **Добавление**. Если имена полей двух таблиц отличны друг от друга, нужно будет указать в строке **Добавление** имена полей таблицы-получателя.

* Для полей, перемещенных в бланк запроса, введите в ячейку **Условие отбора** условие отбора, по которому будет осуществляться добавление. Нашим условием является наличие товара на складе, т.е. в поле НаСкладе в строке Условие отбора введем условие >0
* Для предварительного просмотра добавляемых записей нажмите на панели инструментов в группе Результаты, кнопку **Режим и** выберите режим таблицы. Для возврата в режим конструктора снова нажмите кнопку **Режим** на панели инструментов.
* Для добавления записей нажмите кнопку **Запуск** на панели инструментов. Будет выведено сообщение о добавлении записей в таблицу Товары
* Просмотрите таблицу Товары

Лабораторная работа 5 Разработка форм и отчетов в MS Access

*Цель работы*: Освоение основных приемов работы с формами и отчетами в среде MS ACCESS.

Отчет — это гибкое и эффективное средство для организации данных при выводе на печать. С помощью отчета имеется возможность вывести необходимые сведения в том виде, в котором требуется. Больше всего сведений в отчете берется из базовой таблицы, запроса или инструкции SQL, являющихся источниками данных для отчета. Другие сведения вводятся при разработке отчета.

***Разделы отчета***

Вся информация в отчете разбивается на разделы, каждый из которых имеет специальное назначение. При печати разделы располагаются на страницах в определенном порядке:

1. Заголовок отчета
2. Верхний колонтитул
3. Область данных (данные из таблиц)
4. Нижний колонтитул
5. Примечание отчета

В режиме конструктора на экране отображается макет каждого раздела отчета в одном экземпляре. При печати некоторые разделы могут неоднократно повторяться. Элементы управления, такие как надпись или поле, находящиеся в разделе, определяют местоположение информации в отчете.

Отчет с группировкой данных позволяет вычислить итоговые значения для групп, а также представить информацию в удобном для использования виде..

**Задание**

Используется учебная база данных Борей.

1.Создание детального отчета с помощью мастера отчетов.

Создайте отчет с указанием поставщиков определенных товаров. Необходимая информация берется из запроса Товары по поставщикам.

* Создайте с помощью мастера простой запрос «Товары по поставщикам» на базе таблиц Товары и Поставщики. Включите в него поле Название (т.е. название поставщика) из таблицы Поставщики и все поля таблицы Товары.
* Создайте детальный отчет на базе этого запроса, воспользовавшись мастером отчетов.
* Включите в отчет все поля запроса, выберите тип представления данных по «Поставщики», выберите порядок сортировки по полю Код типа, выберите ступенчатый макет и альбомную ориентацию страницы.
* Назовите отчет «Товары по поставщикам».
* Откройте отчет в режиме конструктора и улучшите расположение заголовков столбцов с данными.

2. Создание суммарного отчета по сгруппированным данным с помощью мастера отчетов.

Выведите суммарные данные продаж по каждому виду товара.

* Используйте в качестве базового готовый запрос «Продажи по типам» в базе данных Борей.
* Добавьте уровень группировки по полю Код типа.
* Не выбирайте полей сортировки данных, а щелкните мышью по кнопке Итоги в диалоговом окне
* Укажите, что по полю Продажи товара необходимо выполнить операцию суммирования, выберите переключатель «данные и итоги».
* Выберите по своему усмотрению макет и стиль отчета
* Дайте название отчету “Продажи товаров по категориям”
* В режиме конструктора удалите ненужные заголовки полей, если таковые имеются.

1. Добавление в отчет текстового поля и изменение порядка сортировки

Необходимо ввести в созданный только что отчет новое текстовое поле и изменить порядок сортировки данных в группе.

* В режиме Конструктора воспользуйтесь элементом Поле на панели элементов для создания текстового поля левее поля с указанием суммы продаж и введите выражение:

=”сумма продаж”&[Категория]

* Раздвиньте текстовое поле и измените начертание шрифта на полужирное.
* В диалоге Сортировка и группировка из меню Вид замените в поле сортировки Марка на сортировку по возрастанию.

1. Удаление разрывов групп и введение нового вида нумерации страниц.

Распечатайте группы без разрывов и введите новый вид нумерации страниц.

* В окне Сортировка и группировка(меню Вид) выделите поле Категория, после чего в окне свойств поместите в графу свойства Не разрывать значение – Полную группу.
* В нижнем колонтитуле удалите текстовое поле, которое содержит выражение для номера страницы.
* Создайте новое тестовое поле, где номер страницы ставится по середине листа после слова страница. Для этого надо всего лишь выполнить команду Вставка/ Номера страниц.

5. Создание отчета в режиме конструктора

Повторите создание отчета “Товары по поставщикам” (см.п.1) в режиме конструктора.

* На вкладке Отчеты выберите команду Создать, затем режим Конструктор. В качестве источника данных укажите запрос Товары по поставщикам
* В открывшемся окне конструктора отчетов найдите все области отчета и окно с полями для отчета
* Если не отображается область заголовка отчета, выполните команду Заголовок/Примечание отчета из верхнего меню Вид
* Создайте Заголовок отчета с помощью элемента Надпись.
* Перетащите имена поля в область данных отчета с помощью мыши. При перетаскивании будет появляться два “прямоугольника” для каждого поля. Один является заголовком(именем поля) и его надо будет перетянуть в область верхнего колонтитула, а другой – значением и он остается в области данных. Если имя поля не удаётся перетянуть в область верхнего колонтитула, удалите его из области данных клавишей Delete и в области верхнего колонтитула с помощью объекта Label создайте заголовок для данных поля вручную.
* Используя кнопку сортировка и группировка, создайте группировку по полю Поставщики
* Сохраните отчет

1. С использованием диспетчера кнопочных форм создать стартовую кнопочную форму, состоящую из 4 страниц: Таблицы, Запросы, Формы, Отчеты. Каждая страница должна содержать 3 кнопки – 2, открывающие любой соответствующий объект, и 1 – возврат на главную страницу.

Главная страница стартовой кнопочной формы должна содержать 5 кнопок – 4 для перехода на страницы с объектами и 1 – выход из приложения.

Для открытия таблиц, запросов и выхода из приложения создать макросы.

Лабораторная работа 6 Базы данных в MS Excel. Фильтры

*Цель работы*: Освоение основных приемов работы с базами данных в среде MS Excel.

Обычная двухмерная таблица Excel, созданная с соблюдением некоторых определенных правил является таблицей базы данных (списком).

Столбцы таблицы Excel – это поля базы данных, а строки – это записи базы данных.

Поле (столбец) содержит информацию об одном признаке для всех записей базы данных. Запись (строка) состоит из нескольких (по числу полей) разнообразных информационных сообщений (признаков), характеризующих один объект базы данных.

Создавая таблицу базы данных в Excel, необходимо выполнять несколько важных правил:

1. Первая строка базы данных должна содержать заголовки столбцов.

2. Каждая последующая строка базы данных должна содержать хотя бы одну заполненную данными ячейку.

3. Не следует применять объединенные ячейки в пространстве таблицы базы данных

4**.**Каждый столбец должен содержать только один тип данных – или текст, или числа, или даты.

5. Вокруг базы данных ячейки должны быть пустыми, либо это должен быть край рабочего пространства листа.

6. Следует отформатировать диапазон базы данных как таблицу, выделив ее и выполнив команду Главная – Стили - Отформатировать как таблицу.

**Сортировка списка**

Для *простой* сортировки данных списка по некоторому столбцу необходимо выделить ячейку внутри сортируемого столбца. Затем на вкладке **Данные** → группе **Сортировка и фильтр** → нажать кнопку **Сортировка по возрастанию**или **Сортировка по убыванию**. Кнопки сортировки доступны также и на вкладке **Главная** в группе **Редактирование***.*

Для *сложной* сортировки (по нескольким столбцам) необходимо выделить ячейку внутри сортируемого списка. Затем выбрать на вкладке **Данные *→*** группе**Сортировка и фильтр *→* Сортировка*.*** MS Excel отобразит диалоговое окно **Сортировка**. Определив критерии сортировки для одного столбца, при необходимости нажать кнопку **Добавить уровень**, выберите имя другого столбца в поле *Затем по* и установите требуемые признак и порядок сортировки. Если первая строка вашего списка не содержит заголовков, выберите в поле **Сортировать по**имя столбца.

**Фильтрация списка**

В Excel имеется два способа отображения записей списка, удовлетворяющих заданным условиям: автофильтр и расширенный фильтр. После выбора на вкладке **Данные** инструмента **Фильтр** в строке имен полей появляются кнопки раскрывающихся списков, содержащих команды *Фильтр по цвету*, *Числовые фильтры, Текстовые фильтры, Фильтры по дате*, а также перечень всех имеющихся значений поля.

Команды **Числовые фильтры, Текстовые фильтры, Фильтры по дате** имеют подкоманды, позволяющие получить значения, соответствующие некоторым условиям. Для отображения всех записей списка надо снова на вкладке **Данные** выбрать инструмент **Фильтр**.

Расширенный фильтр используется для задания сложных условий фильтрации. Чтобы его применить, нужно сначала создать диапазон критериев. Первая строка этого диапазона должна содержать имена полей, по которым будут задаваться условия, в следующих строках вводятся условия. Между критериями в одной строке идет связь по И, между строками критериев — по ИЛИ.

После создания диапазона условий указывается ячейка списка и выбирается на вкладке **Данные** инструмент **Расширенный фильтр**. В этом диалоге **Исходный диапазон** — это диапазон списка. В строке **Диапазон условий** нужно указать ссылку на созданный диапазон критериев. С помощью опций список можно фильтровать на месте или скопировать результат в другое место, указав при этом диапазон из одной строки и стольких столбцов, сколько полей в списке. Начиная с этой строки, будет выведен отфильтрованный список. В расширенном фильтре для задания условий можно также использовать символы шаблонов (\* — заменяет любое количество символов; ? — заменяет один символ) и формулы. Используемая в условии формула должна ссылаться на ячейку первой записи списка в поле, для которого задается условие. В диапазоне критериев с использованием формулы ячейку первой строки, в которой должно быть имя поля, нужно оставить пустой.

**Задание**

1. Сформировать базу данных "Квартирный рынок" со следующими  
полями.

Адрес,

Кол-во этажей в доме,

Этаж,

Кол-во комнат в квартире,

Общая площадь,

Жилая площадь,

Площадь кухни,

Тип дома (панельный, кирпичный, деревянный, 93 серии и др.)

Агентство недвижимости,

Стоимость квартиры,

Дата постройки,

Дата продажи.

База данных должна состоять не менее чем из 30 строк.

2. Отсортировать базу данных

- по типу дома и дате постройки;

- по дате продаже и количеству комнат.

3. Вычислить:

- Среднюю жилую площадь квартир с одинаковым кол-вом комнат,

- Возраст каждой квартиры на текущую дату;

- Выручку каждого агентства недвижимости на каждый день продажи,

- Общее количество квартир, выставленных на продажу каждым агентством недвижимости (с разбивкой квартир по кол-ву комнат),

- Общее кол-во квартир с одинаковым количеством комнат,

- Минимальную стоимость среди квартир с одинаковым количеством  
комнат.

- Среднюю стоимость среди квартир одного типа дома с одинаковым  
количеством комнат.

Результат выполнения каждого задания скопировать на отдельный лист.

3.Выполнить выборку по трехкомнатным квартирам со стоимостью в заданном диапазоне и скопировать ее на отдельный лист.

Лабораторная работа 7 Консолидация данных. Сводные таблицы и сводные диаграммы

*Цель работы*: изучить способы группировки данных и подсчета итоговых значений.

Если в разных листах книги или даже разных книгах имеются однотипные значения, которые нужно разместить в одном месте или провести с ними однотипные операции, это можно сделать при помощи функции **Консолидация**. При ее использовании данные из указанных источников собираются в одном месте, и любые общие значения будут объединяться, как указано. Это позволяет более наглядно анализировать итоговые данные.

Чтобы выполнить консолидацию, сначала нужно выделить первую ячейку места, в котором будут расположены консолидированные данные, затем выбрать команду **Данные** – **Работа с данными** – **Консолидация**. В диалоговом окне необходимо выбрать диапазоны ячеек (поля ссылка, список диапазонов) и тип предстоящей консолидации (поле Функция), а также указать использование в качестве имен некоторых значений диапазона данных и установить связь с исходными данными (чтобы при изменении исходных данных производилось обновление консолидированных). После задания параметров итоговые данные будут размещены в указанных ячейках.

При консолидации *по расположению* Excel применяет итоговую функцию(Сумма, Среднее или другую) к ячейкам с одинаковыми адресами во всех исходных листах. Это простейший способ консолидации, при котором консолидируемые данные во всех исходных листах должны иметь совершенно одинаковое расположение.

Консолидация *по категории* используется, если исходные листы содержат различное количество строк, но одинаковое количество столбцов

**Сводные таблицы** обеспечивают очень удобный интерфейс к хранилищам данных различной сложности и разного объема. Сводная таблица – это динамическая таблица специального вида, построенная на базе одной или нескольких исходных таблиц и содержащая сводную информацию по этим таблицам. Базами данных для сводных таблиц могут быть списки, таблицы, расположенные на рабочих листах Excel, либо внешние источники данных (например, базы данных Access). При создании сводной таблицы пользователь распределяет информацию, указывая, какие элементы и в каких полях сводной таблицы будут содержаться. **Поле** – это некоторая совокупность данных, собранных по одному признаку. **Элемент** – отдельное значение, содержащееся в поле.

Сводная таблица является многомерной и всегда связана с источником данных. Сама она предназначена только для чтения, а *изменения нужно вносить в исходные таблицы*. При этом можно изменять форматирование сводной таблицы, выбирать различные параметры вычислений.

Для работы в Excel со сводными таблицами существует команда **Вставка** – **Таблицы** – **Сводная таблица**. После ее активизации в появившемся окне **Создание сводной таблицы** нужно указать исходные данные и размещение итогов сводной таблицы.

После этого при помощи мастера **Список полей сводной таблицы** необходимо заполнить макет таблицы, для чего выберите и перетащите имена нужных полей в области **Названия строк**, **Названия столбцов**, **Фильтр отчета** или **Значения.**

После того как поля назначены областям, можно изменить тип обобщенной информации (счет, суммирование, среднее значение и т.д.) каждого из них, для чего необходимо дважды щелкнуть на поле. Так же можно изменить название самого поля.

В результате будет получена сводная таблица, после вставки которой, на ленте меню появляется контекстный инструмент **Работа со сводными таблицами**, имеющий вкладки **Параметры** – для изменения структуры сводной таблицы, и **Конструктор** – для ее форматирования.

**Задание**

1. В магазине имеется 5 видов товаров (папки, скрепки, кнопки, скоросшиватели, степлеры). Продажа этих товаров осуществлялась в течение двух кварталов (1 квартал - январь, февраль, март; 2 квартал - апрель, май, июнь). Для оценки суммы выручки поквартально, необходимо создать на трех листах таблицы с данными: 1 лист - 1квартал; 2лист -2квартал; З лист - итого). Воспользовавшись командой Консолидация сделать эти таблицы связанными и в итоговой таблице подсчитать сумму выручки по каждому товару за каждый квартал.

2. Для базы данных "Квартирный рынок" из предыдущей лабораторной работы создать на отдельных листах:

- сводную таблицу по средней стоимости квартир различных типов с одинаковым количеством комнат.

- сводную таблицу о продаже квартир по датам.

Лабораторная работа 8 Ввод-вывод данных. Разработка программы линейной структуры

*Цель работы*: изучение *Visual Basic for Application* на примере линейной программы изучение организации ввода и вывода данных с помощью диалоговых окон, и с помощью ячеек листа Excel.

***Для ввода данных с листа или вывода на лист*** используется объект Worksheets и его методы Range или Cells.

Метод Range использует в качестве аргументов одну или две ссылки на ячейки и возвращают объект Range. Ссылки на ячейки должны быть оформлены в стиле А1 (колонка-буква-строка-число). Например, выражение

X = Worksheets(“Лист1”).Range(“B1”).Value - присваивает переменной Х значение ячейки «B1» листа «Лист1»,

Worksheets(“Лист1”).Range(“В7:С9”).Value = 3 - выводит в диапазон ячеек «В7:С9» листа «Лист1» число 3.

Метод Cells, получая в качестве аргументов два целых числа, возвращают объект, содержащий единичную ячейку. Аргументы определяют номера строки и столбца выбранной ячейки. Например, выражение

A=Worksheets(1).Cells(1,2).Value - переменной А присвоено значение из ячейки первой строки и второго столбца первого листа,

Worksheets(1).Cells(2,2).Value= Х - в ячейку второй строки и второго столбца заносится значение переменной Х.

***Для ввода данных с клавиатуры*** используется окно ввода **InputBox**, а ***для вывода информации на экран*** – окно сообщений **MsgBox**.

Функция **InputBox** выводит на экран диалоговое окно, содержащее сообщение, поле ввода и две кнопки OK и Cancel. Устанавливает режим ожидания ввода текста пользователем или нажатия кнопки, а затем возвращает значение типа string по нажатию кнопки OK, содержащее текст, введенный в поле. При нажатии кнопки Cancel возвращает пустую строку.

Синтаксис: InputBox(prompt[, title] [, default])

**prompt** — строковое выражение, отображаемое как сообщение в диалоговом окне. Строковое значение prompt может содержать несколько строк. Для разделения строк допускается использование символа возврата каретки (chr (13)), символа перевода строки (chr(10)) или комбинацию этих символов (chr( 13) & Chr (10));

**title** — строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот аргумент опущен, в строку заголовка помещается имя приложения;

**default** — строковое выражение, отображаемое в поле ввода как используемое по умолчанию, если пользователь не введет другую строку. Если этот аргумент опущен, поле ввода изображается пустым.

***Пример***.

Имя =InputBox(“Введите Ваше имя”, “Пример окна ввода”)

На экране появится окно, представленное на рисунке 8.1.

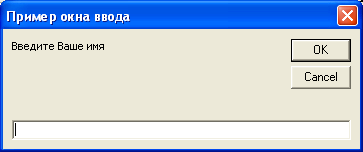


Рисунок 8.1 – Пример окна ввода

Переменной Имя будет присвоено значение типа String, введенное пользователем.

Процедура **MsgBox** выводит на экран диалоговое окно, содержащее сообщение, устанавливает режим ожидания нажатия кнопки пользователем, а затем возвращает значение типа integer, указывающее, какая кнопка была нажата. Синтаксис:

**MsgBox**(prompt[, buttons] [, title])

Аргументы:

**prompt** — строковое выражение, отображаемое как сообщение в диалоговом окне;

**buttons** — числовое выражение, представляющее сумму значений, которые указывают число и тип отображаемых кнопок, тип используемого значка, основную кнопку и модальность окна сообщения. Значение по умолчанию этого аргумента равняется 0. Значения констант, определяющих число и тип кнопок используемого значка, приведены в таблицах 5-6;

**title** — строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот аргумент опущен, в строку заголовка помещается имя приложения

*Пример*.

N = MsgBox (“Значение переменной Х=” & X & Chr(10) & “Продолжить вычисления?”, VbYesNo, “Пример окна MsgBox”)

Если к моменту выполнения данного оператора переменная Х равнялась числу 2,14587895, то на экране появится окно, представленное на рисунке 8.2.

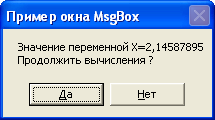


Рисунок 8.2 – Пример окна сообщений

Часто процедура MsgBox используется в «минимальном» варианте - только для вывода сообщения, с одной кнопкой – OK. В этом случае аргументы не берутся в скобки. Например: MsgBox “Значение переменной Х=” & X

**Задание**

Составьте блок-схему и программу на языке VBA для вычисления выражения и решения задачи. Ввод-вывод данных выполнить двумя способами: с использованием рабочего листа Excek и диалоговых окон ввода-вывода. В диалоговых окнах вывода использовать значки и дополнительные кнопки.

|  |  |  |
| --- | --- | --- |
| № варианта | Задание 1 | Задание 2 |
| 1. |  | Вычислить объема прямоугольного параллелепипеда. |
| 2. |  | Вычислить расстояние между населенными пунктами, изображенными на карте.  Исходные данные:  масштаб карты (количество километров в одном см), расстояние между точками изображающими населенные пункты (см). |
| 3. |  | Вычислить площадь треугольника, если известна длина основания и высота. |
| 4. |  | Вычислить стоимость поездки на автомобиле на дачу (туда и обратно).  Исходными данными являются: расстояние до дачи (в километрах); количество бензина, которое потребляет автомобиль на 100 км пробега; цена одного литра бензина. |
| 5. |  | Вычислить объем куба. |
| 6. |  | Пересчитать расстояние из верст в километры (1 верста равняется 1 066,8 м). |
| 7. |  | Вычислить объем цилиндра. |
| 8. |  | Вычислить стоимость некоторого количества (по весу) яблок, если известна цена 1 кг. |
| 9. |  | Вычислить стоимость покупки, состоящей из нескольких тетрадей и такого же количества обложек к ним.  Исходные данные: цена тетради, цена обложки, количество комплектов. |
| 10. |  | Вычислить площадь треугольника, если известны длины двух его сторон и величина угла между этими сторонами. |
| 11. |  | Пересчитать вес из фунтов в килограммы(1 фунт равняется 409,5 г.). |
| 12. |  | Вычислить площадь прямоугольного треугольника, если известны длины двух его гипотенуз. |
| 13. |  | Вычислить длину диагонали прямоугольника |
| 14. |  | Вычислить стоимость некоторого количества (по весу) капусты, если известна цена 1 кг. |
| 15. |  | Вычислить площадь треугольника, если известна длина основания и высота. |

Лабораторная работа 9 Разработка программы разветвленной структуры

*Цель работы*: Изучение оператора условия IF.

Оператор ветвления позволяет выбирать и выполнять действия в зависимости от истинности некоторого условия. Имеется два варианта синтаксиса:

1) *однострочная* форма записи условного оператора:

**IF** условие **Then [**операторы1**] [Else** операторы2**]**

2) *блочная* форма записи (оператор **IF** расположен на нескольких строках и может проверять несколько условий):

**IF** условие 1 **Then**

**[**операторы 1**]**

**[ElseIf** условие **–** n **Then**

**[**операторы**-**n **]…**

**[Else**

**[**ИначеОператоры**]]**

**End If**

Здесь условие обязательно в обоих вариантах. Оно может быть числовым или строковым выражением со значениями TRUE или FALSE. Если условие истинно (TRUE), выполняется последовательность “операторы 1”, если ложно, “операторы 2”.

Изображение в блок-схеме:

нет

да

Х>0

условие

истинно

Операторы1

Операторы2

**Задание**

Составьте блок-схему и программу на языке VBA для выполнения заданий.

|  |  |
| --- | --- |
| Номер варианта | Задание |
| **1** | Вычислить:  **1.**  **2**. Ввести три целых числа и определить, сумма каких двух является наибольшей. Числа вводить, используя InputBox. |
| 2 | Вычислить:  **1.**  **2** Ввести число Х и определить, делится ли оно:   1. на 2, 2. на 3, 3. на 5.   Результаты проверок вывести. Числа вводить, используя InputBox. |
| 3 | Вычислить:  **1.**  2 Ввести число Х и определить, делится ли оно на 8 без остатка. Число вводить, используя InputBox. |
| 4 | Вычислить:  **1**  2 . Ввести число Х и определить, является ли оно четным. |
| 5 | Вычислить:  **1.**  **2.** Ввести число N и определить делится ли оно без остатка на число M. Числа вводить, используя InputBox. |
| **6** | 1. Вычислить значение функции:  sin(x), если х>0,  1 f(x) = 0, если х = 0,  сos(x), если х< 0.  2. Ввести три целых числа и определить, произведение каких двух является максимальным. Числа вводить, используя InputBox. |
| **7** | Вычислить:  **1.**  **2.** Ввести число Х и определить, делится ли оно:  - на 2,  - на 3,  - на 5. |
| **8** | 1. Сравнить, что больше Sin(x), Tg(x) или Cos(x)  **2.** |
| **9** | 1. Составьте программу для нахождения максимального из 3-х введенных чисел.  2 Составьте программу, которая запрашивает два числа, запрашивает ответ на сумму этих чисел, проверяет его и выводит сообщение «Правильно» или «Вы ошиблись» и правильный результат. |
| **10** | 1. Составьте программу, которая проверяет, является ли введенные три числа углами треугольника. (ввод в градусах)  2. |
| **11** | **1.**  если *x* = sin *b* + cos *d*.  2. Составьте программу, которая запрашивает два числа, запрашивает ответ на сумму этих чисел, проверяет его и выводит сообщение: «Правильно» или «Вы ошиблись» и правильный ответ. |
| **12** | 1. Вычислить значение функции:  sin(x), если х>0,  1 f(x) = 0, если х = 0,  сos(x), если х< 0.  2. Ввести три целых числа и определить, произведение каких двух является максимальным. Числа вводить, используя InputBox. |
| **13** | Вычислить:  **1.**  2 Ввести число Х и определить, делится ли оно на 8 без остатка. Число вводить, используя InputBox. |
| **14** | Вычислить:  **1.**  **2**. Ввести три целых числа и определить, сумма каких двух является наибольшей. Числа вводить, используя InputBox. |
| **15** | 1. Составьте программу, которая проверяет, является ли введенные три числа углами треугольника. (ввод в градусах)  2. |

Лабораторная работа 10 Оператор цикла с параметром For…Next. Обработка массивов

*Цель работы*: Изучение оператора цикла с параметром For…Next. Освоение основных способов обработки массивов.

Оператор цикла For позволяет повторять группу операторов заданное число раз. Синтаксис:

**For** *счётчик\_цикла* = *начало* **to** *конец* [step шаг]

*Тело цикла*

**Next** [*счётчик\_цикла*]

Здесь ***счётчик\_цикла*** – это числовая переменная. В начале выполнения цикла она принимает значение, задаваемое числовым выражением ***начало.*** Числовое выражение ***конец –*** задает заключительное выражение счётчика цикла. Числовое выражение шаг не обязательно и по умолчанию=1. ***Тело цикла –*** это последовательность операторов которая будет выполнена заданное число раз. Если шаг положителен, цикл завершится, когда впервые выполнится условие**:**

***счетчик\_цикла > конец***

Если шаг цикла отрицателен, условие его завершения:

***счетчик\_цикла < конец***

Это условие проверяется перед началом выполнения цикла, а затем – после каждого прибавления шага к счётчику цикла в операторе Next. Если оно выполнено, управление передается на оператор, следующий за Next, нет – выполняются операторы из тела цикла. Завершить цикл For…Next можно и с помощью оператора Exit For. Такие операторы могут быть расположены в тех местах тела цикла, где требуется из него выйти не дожидаясь выполнения условия завершения.

Массив – совокупность однотипных элементов данных (чисел, логических данных, символов), которой при обработке присвоено определенное имя. Массивы бывают статические и динамические. **Статическими** называются массивы, количество элементов в которых заранее известно и не изменяется в ходе выполнения программы. **Динамические** массивы – массивы, в которых либо не известно начальное количество элементов, либо размерность массива (количество элементов) изменяется при выполнении программы.

Описание массивов:

1) **одномерный статический массив**

**Dim** *<имя массива>* **(** *<начальное значение индекса*> **To** *<конечное значение индекса>***)** [**As** *<тип элементов массива*>]

или

**Dim** *<имя массива>* **(** *<количество элементов массива*>**)** [**As** *<тип элементов массива*>];

2) **двумерный статический массив**

**Dim** *<имя массива*> **(** *<начальное значение индекса по строкам>* **To** *<конечное значение индекса по строкам* >**,** *< начальное значение индекса по столбцам*> **To** *< конечное значение индекса по столбцам>***)** [**As** *<тип элементов массива>*]

или

**Dim** <имя массива> **(** <количество строк>**,** <количество столбцов>**)** [**As** <тип элементов массива>].

Первый способ отличается от второго тем, что в первом случае указывается индекс первого и последнего элементов, во втором же – только количество элементов, нумерация которых может начинаться как с 0, так и с 1. Это зависит от опции **Base** (задает базовый индекс). Если опция не указана, то нумерация элементов массива начинается с нуля. Для изменения базового индекса в начале листа модуля необходимо написать **Option Base 1**.

3) **динамический массив:**

**Dim** <имя массива> **( )** [**As** <тип элементов массива>].

После определения количества элементов массива выполняется его переопределение:

**ReDim** <имя массива> **(**<задается размерность массива (одномерного/двумерного >**)**.

Обращение к элементу массива осуществляется следующим образом: указывается имя массива, а затем в круглых скобках указывается номер элемента в массиве. Если массив двумерный – указывается вначале номер строки, затем через запятую номер столбца.

**Задание**

Составьте блок-схему и программу на языке VBA для выполнения заданий.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вариант | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| № задания | 1, 24 | 2, 23 | 3, 22 | 4, 21 | 5, 20 | 6, 19 | 7, 18 | 8, 17 | 9, 16 | 10,15 | 11, 14 | 12, 13 | 5, 20 | 2, 18 | 4, 15 |

1. Создать одномерный массив из 20 целочисленных значений. Найти наибольшее среди положительных элементов, стоящих на четных местах, и вывести его значение и индекс данного элемента.
2. Создать одномерный целочисленный массив из 20 значений. Подсчитать среднее арифметическое значение всех элементов и вывести на экран все значения, меньшие среднего арифметического.
3. Создать одномерный целочисленный массив из 20 значений. Расставить в нем элементы в обратном порядке.
4. Создать одномерный целочисленный массив из 20 значений. Поменять местами каждую пару чисел, например, А1 с А2, А3 с А4 и т.д.
5. Создать одномерный массив из 20 целочисленных значений. Найти минимальную сумму каждой пары значений, т.е. С1 и С2, С3 и С4.
6. Создать два одномерных массива по 10 целочисленных значений каждый. Сформировать третий массив из 20 элементов следующим образом: на нечетные места ставить элементы из первого массива, на четные места – из второго.
7. Создавать одномерный массив из 20 целочисленных значений. Элементы, стоящие на нечетных местах, расставить в обратном порядке.
8. Создать два одномерных массива по 20 целочисленных значений. Совершить обмен данными: в одном массиве с 1-ого по 10- ый элемент, в другом с 11 – го по 20 – ый элемент, т.е. Х1 на Y11, Y11 на X1 и т.д.
9. Создать одномерный массив из 20 целочисленных значений. Расставить в нем в том же порядке сначала положительные элементы, затем отрицательные, затем нулевые.
10. Создать одномерный массив из 20 целочисленных значений. Найти минимальное среди них, вывести его и индекс данного элемента на экран.
11. Создать одномерный массив из 20 целочисленных значений. Найти минимальное и максимальное значения, вывести их разность на экран.
12. Создать одномерный массив из 20 целочисленных значений. Найти наибольшее среди отрицательных и вывести его индекс на экран.
13. Дан массив размерностью n x m. Определить количество строк, содержащих хотя бы один нулевой элемент.
14. Дан массив размерностью n x m. Заменить все положительные элементы на 1, все отрицательные на -1.
15. Дан массив размерностью n x m. Подсчитать сумму элементов, у которых сумма номера строки и номера столбца равна n=1.
16. Дан массив размерностью n x m. Подсчитать сумму элементов, у которых модуль разности номера строки столбца равен 1.
17. Дан массив размерностью n x m. Подсчитать сумму элементов, которые меньше номера своей строки.
18. Дан массив размерностью n x m. Подсчитать сумму элементов, которые больше номера своего столбца и строки.
19. Дан массив размерностью n x m. Подсчитать сумму положительных элементов и распечатать их номера.
20. Дан массив размерностью n x m. Подсчитать сумму нечетных элементов и распечатать их номера.
21. Дан массив размерностью n x m. Подсчитать сумму элементов, имеющих одинаковые остатки при делении на 7 и на 2, и распечатать номера таких элементов.
22. Дан массив размерностью n x m. Преобразовать элементы массива по следующему правилу: если элемент четный то разделить его на 2, если нечетный – заменить его остатком от деления на 3.
23. Дан массив размерностью n x m. Преобразовать элементы массива по следующему правилу: если элемент положительный, то умножить его на 2, а ели отрицательный – поменять знак на противоположный.
24. Дан массив размерностью n x m. Преобразовать элементы массива по следующему правилу: если элемент четный, то прибавить к нему 1, если нечетный – умножить на 2.

Лабораторная работа 11 Разработка программ циклической структуры

*Цель работы*: Освоение основных приемов применения операторов цикла с предусловием и постусловием

Циклы данного вида используются, когда заранее неизвестно, сколько раз будет выполняться тело цикла.

Циклы с предусловием (Do While…Loop, While…Wend, Do Until…Loop) представлены в таблице 11.1, а операторы циклов с постусловием (Do…Loop While, Do…Loop Until) - в таблице 11.2.

Отличие циклов с предусловием от циклов с постусловием заключается в том, что тело цикла первых может не выполниться ни разу, в то время как тело цикла с постусловием всегда выполнится хотя бы один раз.

Таблица 11.1

|  |  |  |  |
| --- | --- | --- | --- |
| Синтаксис | **Do Whil**e <*условие*>  <*тело цикла*>  [**Exit Do**]  …  **Loop** | **Whil**e *<условие*>  <*тело цикла>*  **Wend** | **Do Unlil** <*условие>*  *<тело цикла*>  [**Exit Do**]  …  **Loop** |
| Порядок выполнения | ***<****Тело цикла>* будет выполняться в том случае, когда *<условие*> имеет значение истина (TRUE) (цикл продолжается при истинном значении *<условия***>**). Если *<условие*> ложно (FALSE), то выполняются операторы, стоящие за циклом. В первом случае есть возможность досрочного выхода из цикла (это реализовано через **Exit Do)** | | ***<****Тело цикла*> выполняется до тех пор, пока *<условие*> не примет значение истина (цикл продолжается при ложном значении *<условия*>). Есть возможность досрочного выхода из цикла (это реализовано через **Exit Do**) |
| Изображение в блок-схемах | <условие>  <тело цикла>  да  нет | | <тело цикла>  да  нет  <условие> |

Таблица 11.2

|  |  |  |
| --- | --- | --- |
| Синтаксис | **Do**  <*тело цикла*>  [**Exit Do**]  …  **Loop While** *<условие>* | **Do**  *<тело цикла>*  [**Exit Do**]  …  **Loop Until *<****условие>* |
| Порядок выполнения | *<Тело цикла>* будет выполняться в том случае, когда *<условие>* имеет значение истина (цикл продолжается при истинном значении *<условия*>). Если <*условие*> ложно, то выполняются операторы, стоящие за циклом. Предоставлена возможность досрочного выхода из цикла (это реализовано через **Exit Do**) | ***<****Тело цикла>* выполняется до тех пор, пока *<условие>* не примет значение истина (цикл продолжается при ложном значении *<условия*>). Есть возможность досрочного выхода из цикла (это реализовано через **Exit Do**) |
| Изображение в блок-схемах | <условие>  <тело цикла>  да  нет | <условие>  <тело цикла>  да  нет |

**Задание**

Составьте блок-схему и программу на языке VBA для выполнения заданий.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вариант | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| № задания | 1, 8 | 2, 9 | 3, 10 | 4, 11 | 5, 12 | 6, 13 | 7, 14 | 1, 8 | 2, 9 | 3, 10 | 4, 11 | 5, 12 | 6, 13 | 7, 14 | 4, 11 |

1. Начиная с какого номера n имеет место неравенство n!>xn, x - любое действительное число.
2. Вводить последовательность до тех пор, пока не встретятся три подряд идущих положительных числа. Тогда прервать ввод и сообщить, сколько во введенной последовательности было всего чисел.
3. Вводить последовательность до тех пор, пока не встретятся три подряд идущих положительных числа. Тогда прервать ввод и сообщить, сколько во введенной последовательности было положительных чисел.
4. Вводить последовательность до тех пор, пока не встретятся три подряд идущих положительных числа. Тогда прервать ввод и сообщить, сколько во введенной последовательности было отрицательных чисел.
5. Определить сколько натуральных подряд идущих четных чисел нужно сложить (найти минимальное число таких слагаемых), чтобы их сумма была больше введенного числа.
6. Дано натуральное N и первый член бесконечного ряда: Y1=1. Вычислить сумму членов бесконечного ряда, образованного по следующему рекуррентному соотношению: Yi=2\*Yi-1 (то есть S=1+2+4+8+16+...). Вычисление суммы продолжать до тех пор, пока соблюдается условие |Yi-Yi-1|<N.
7. Последовательно вводятся числа до тех пор, пока во введенной совокупности не окажется три нуля. Вывести количество введенных чисел.
8. Дано натуральное число.

а) Верно ли, что сумма его цифр меньше А?

б) Верно ли, что произведение его цифр больше В?

в) Верно ли, что это число k-значное?

1. Дано натуральное число. Определить номер цифры 3 в нем, считая от конца числа. Если такой цифры нет, ответом должно быть число 0, если таких цифр в числе несколько — должен быть определен номер самой правой из них.

10. Дано натуральное число. Если в нем есть цифры "2" и "5", то определить, какая из них расположена в числе левее. Если одна или обе эти цифры встречаются в числе несколько раз, то должны быть рассмотрены самые левые из одинаковых цифр.

11. Дано натуральное число. Определить:

а) количество цифр в нем;

б) сумму его цифр;

в) произведение его цифр;

г) среднее арифметическое его цифр;

12. Дано натуральное число. Определить:

а) сумму квадратов его цифр;

б) сумму кубов его цифр;

в) его первую цифру;

г) сумму его первой и последней цифр.

13. Дано натуральное число.

а) Верно ли, что его первая цифра не превышает 6?

б) Верно ли, что оно начинается и заканчивается одной и той же цифрой?

в) Определить, какая из его цифр больше: первая или последняя.

14. Дано натуральное число. Определить:

а) количество цифр "3" в нем;

б) сколько раз в нем встречается цифра, равная последней;

в) количество четных цифр в нем.

Лабораторная работа 12 Обработка строковых данных

*Цель работы*: Изучение функций обработки строк.

Строка — упорядоченная последовательность символов. Каждый символ строковой величины занимает 1 байт памяти (код АSСII). Количество символов в строке называется ее длиной.

Строковая константа — последовательность символов, заключенных в кавычки. Например: “это строковая константа” , “272”. Две следующих друг за другом кавычки (“”) обозначают “пустую строку”, т.е. строку нулевой длины.

Строковая переменная описывается в разделе описания переменных:

**Dim** <идентификатор> **As String**

Например:

Dim Name As String

**Операции над строками.**

*Операция сцепления* (конкатенации) (&) применяется для соединения нескольких строк в одну результирующую строку. Сцеплять можно как строковые константы, так и переменные.

*Операции отношения*: =, <, >, <=, >=, <> позволяют произвести сравнение двух строк, в результате чего получается логическое значение (true или false). Операция отношения имеет приоритет более низкий, чем операция сцепления. Сравнение строк производится слева направо до первого несовпадающего символа, и та строка считается больше, в которой первый несовпадающий символ имеет больший номер в таблице символьной кодировки. Если строки имеют различную длину, но в общей части символы совпадают, считается, что более короткая строка меньше, чем более длинная. Строки равны, если они полностью совпадают по длине и содержат одни и те же символы.

В VBA имеются следующие функции обработки строковых выражений.

|  |  |  |
| --- | --- | --- |
| Функция | Возвращаемое выражение | |
| Asc | Возвращает ASCII-код начальной буквы строки.  Синтаксис: Asc(Строка) | |
| Chr | Преобразует ASCII-код в строку.  Синтаксис: Chr(Код)  Например Chr (13) — переход на новую строку, Chr(97) = "а" | |
| Lcase | Преобразует строку к нижнему регистру.  Синтаксис: Lcase(Строка) | |
| Ucase | Преобразует строку к верхнему регистру.  Синтаксис: Ucase(Строка) | |
| Left | Возвращает подстроку, состоящую из заданного числа первых символов исходной строки.  Синтаксис: Left(string, length)  Аргументы: string —исходная строка, length — число символов. | |
| Right | Возвращает строку, состоящую из заданного числа последних символов исходной строки.  Синтаксис: Right(string, length)  Аргументы: string — исходная строка, length — число символов | |
| Mid | Возвращает подстроку строки, содержащую указанное число символов.  Синтаксис: Mid(string, start [, length])  Аргументы: string — строковое выражение, из которого извлекается подстрока, start — позиция символа в строке string, с которого на­чинается нужная подстрока, length — число возвращаемых символов подстроки. | |
| Len | Возвращает число символов строки.  Синтаксис: Len(Строка) | |
| LTrim | Возвращает копию строки без пробелов в начале.  Синтаксис: LTrim(Строка) | |
| Rtrim | Возвращает копию строки без пробелов в конце.  Синтаксис: RTrimfСтрока) | |
| Trim | Возвращает копию строки без пробелов в начале и в конце  Синтаксис: Trim(Строка) | |
| Space | Возвращает строку, состоящую из указанного числа пробелов.  Синтаксис: Space(Число) | |
| String | Возвращает строку, состоящую из указанного числа повторений одного и того же символа.  Синтаксис: String(number,character)  Аргументы: number — число повторений символа, character — повторяемый символ | |
| InStr | Возвращает позицию первого вхождения одной строки внутри другой строки.  Синтаксис:  InStr([start, ]stringl, string2[, compare])  Аргументы: start — числовое выражение, задающее позицию, с кото­рой начинается каждый поиск. Если этот аргумент опущен, поиск начинается с первого символа строки, stringl — строковое выражение, в котором выполняется поиск, string2 — искомое строковое выражение, compare — указывает способ сравнения строк. Допустимые значения: 0 (для двоичного сравнения), 1 (посимвольное сравнение без учета регистра). | |
| Val (строка) | | | Возвращает числа, содержащиеся в строке, как числовое значение соответствующего типа |
| Str (число) | | | Возвращает значение типа variant (string), являющееся строковым представлением числа |

**Задание**

Составьте блок-схему и программу на языке VBA для выполнения заданий.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вариант | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| № задания | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

1. Заменить в тексте букву “о” на цифру “0”, а букву “з” на цифру “3”. Вывести количество этих замен.
2. Определить, является ли введенное 6-значное число числом-палиндромом. Числа-палиндромы — это такие числа, которые читаются одинаково как справа налево, так и слева направо, т.е. 654456, 112211 и т.д.
3. Задан текст. Проверить, какая из букв «а» или «е» встречается чаще.
4. Задан текст. Распечатать номера позиций, на которых встречается буква «к».
5. Определить, сколько в тексте слов заканчивается на букву «о».
6. Известно, что в заданный текст входит хотя бы одна буква «а», причем не на последнем месте. Напечатать по одной литере текста, следующей непосредственно за буквой «а».
7. Определить, сколько раз в тексте встречается введенное слово.
8. Определить, какой процент слов в тексте содержит хотя бы одну букву “d”.
9. Напечатать те слова из текста, в которых нет удвоенной буквы.
10. Удалить из текста все удвоенные буквы.
11. Из заданного текста удалить последовательность символов между двумя двоеточиями (хотя бы одна пара двоеточий имеется).
12. Перед каждой цифрой в тексте вставить знак «+».
13. Напечатать самое длинное слово из заданного текста.
14. Напечатать удвоенные буквы в строке и их позиции.
15. Определить, сколько в тексте слов заканчивается на букву «е».

Лабораторная работа 13 Разработка программ с использованием подпрограмм и функций

*Цель работы:* разработка программ с использованием подпрограмм и функций.

*Подпрограмма*— программа, реализующая вспомогательный алгоритм. *Основная программа —* программа, реализующая основной алгоритм решения задачи и со­держащая в себе обращения к подпрограммам. В VBA существуют два типа подпрограмм:

**- подпрограммы-функции**

**- подпрограммы-процедуры**.

*Отличие* функции от процедуры заключается  в том, что результатом исполнения операторов, образующих тело функции, всегда является некоторое единственное значение, поэтому обращение к функции можно использовать в соответствующих выражениях наряду с переменными и константами.

**Подпрограмма-процедура** имеет следующий синтаксис:

**[Private | Public] Sub <имя процедуры> ([<список аргументов>])**

**[<Инструкции>]**

**[Exit Sub]**

**[<Инструкции>]**

**End Sub**

**Public –** указывает, что процедура Sub доступна во всех других процедурах во всех модулях;

**Private –** указывает, что процедура Sub доступна для других процедур только для того модуля, в котором она описана;

**Sub, End Sub –** служебные слова VBA;

**<имя процедуры> –** имя процедуры, удовлетворяющее стандартным правилам именования;

**<список аргументов> –** список переменных, представляющих аргументы, которые передаются в процедуру Sub при её вызове. Имена переменных разделяются запятой;

**<Инструкции> –** любой набор команд VBA;

**Exit Sub –** инструкция, выполнение которой приводит к выходу из процедуры.

Синтаксис элемента **<список аргументов>**:

**[ByVal | ByRef] <Имя переменной> [As <Тип>]**

**ByVal –** ключевое слово, указывающее, что аргумент передается по значению;

**ByRef –** ключевое слово, указывающее, что аргумент передается по ссылке. Описание **ByRef** используется в VBA по умолчанию;

**<Имя переменной> –** имя переменной, удовлетворяющее стандартным правилам именования переменных;

**<Тип> –** тип данных аргумента, переданного в процедуру;

В качестве результата процедура может возвращать в вызывающую программу множество простых или структурированных величин или не возвращать никаких значений. Среди параметров процедуры указываются как аргументы, так и ре­зультаты.

Обращение к процедуре — отдельный оператор.

Вызов процедуры из другой процедуры можно произвести несколькими способами.

Первый способ:

**<Имя процедуры> <Список фактических параметров>**

**<Имя процедуры> –** имя вызываемой процедуры;

**<Список фактических параметров> –** список аргументов, передаваемых процедуре; он должен соответствовать списку, заданному в процедуре, по количеству и типу.

Второй способ:

**Call <Имя** процедуры**> (<Список фактических параметров>)**

**Call –** служебное слово VBA;

**<Имя процедуры> –** имя вызываемой процедуры;

**<Список фактических параметров> –** список аргументов, передаваемых процедуре; он должен соответствовать списку, заданному в процедуре, по количеству и типу.

Заметим, что при втором способе вызова процедуры в отличие от первого список фактических параметров должен быть заключен в круглые скобки; в качестве разделителя в списке используется запятая.

**Подпрограмма-функция** имеет следующий синтаксис:

**[Private | Public] Function <имя функции> [(<список аргументов>)]**

**[<Инструкции>]**

**[Exit Function]**

**[<Инструкции>]**

**End Sub**

Синтаксис инструкции Functionсодержит те же элементы, что и Sub.

Тип функции может быть только простым типом. Для возврата значения из функции следует присвоить значение имени функции.

Обращение к функции является операндом в выражении. Подпрограмма-функция вызывается в выражении по своему имени, за которым следует вписок аргументов в скобках.

При обращении к подпрограмме происходит *передача* ей *аргументов по ссылке* (если формальный параметр является параметром-переменной, описан как ByRef) или *по значению* (является параметром-значением, описан как ByVal).

Если параметр определен как *параметр-значение* (с помощью ключевого слова ByVal), то перед вызовом подпрограммы это значение вычисляется, полученный результат копируется во временную память и передается подпрограмме. Любые возможные изменения в подпрограмме параметра-значения никак не воспринимаются вызывающей подпрограммой, так как в этом случае изменяется копия фактического параметра.

Если параметр определен как *параметр-переменная* (по умолчанию или с помощью ключевого слова ByRef), то при вызове подпрограммы передается сама переменная, а не ее копия. Изменение параметра-переменной приводит к изменению самого фактического параметра в вызывающей подпрограмме.

Если в качестве фактического параметра используется константа, транслятор блокирует любые присваивания константе нового значения в теле подпрограммы.

**Задание**

Составьте блок-схему и программу на языке VBA для выполнения заданий. Задание выполнить двумя способами: с использованием подпрограммы-функции и подпрограммы-процедуры.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вариант | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| № задания | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

1. Рассчитать значение выражения :

2. Даны два натуральных числа. Выяснить, в каком из них сумма цифр больше. (Определить функцию и процедуру для расчета суммы цифр натурального числа.)

3. Рассчитать значение выражения :

4. Даны основания и высоты двух равнобедренных трапеций. Найти сумму их периметров и сумму их площадей. (Определить функцию и процедуру для расчета периметра и площади равнобедренной трапеции по ее основаниям и высоте.)

5. Найти значение выражения

где *n!* означает факториал числа *n.* (Определить функцию и процедуру для расчета факториала натурального числа.)

6. Даны две последовательности целых чисел: *a,, ..., ик* и *b]t b,, ..., bу* Найти количество четных чисел в первой из них и количество нечетных во второй. (Определить функцию и процедуру, позволяющую распознавать четные числа.)

7. Рассчитать значение выражения*,* определив функцию и процедуру:

8 Определить значение

Е = min(a, *3b) \** min(2a — b, 2b),

где min(a, b) есть минимальное из чисел a*,b.* Прирешении задачи определить и использовать функцию и процедуру min.

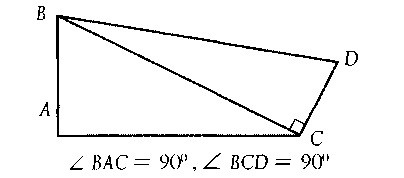
9. Определить значение

Z= max(a, 2b) • max(2a — *b, b),*

где max(x, y) есть максимальное из чисел *x, у.* При решении задачи определить и использовать функцию и процедуру max.

10. Даны три квадратных уравнения *ах2* + *bx* + *с =* 0, *bх2* + *ах* + *с =* 0, *cx2* + *ах + b =* 0. Сколько из них имеют вещественные корни? (Определить функцию и процедуру, позволяющую распознавать наличие вещественных корней в квадратном уравнении.)

11. Найти периметр фигуры *ABCD* по заданным сторонам AB, АС и *DC.* (Определить функцию и процедуру для расчета гипотенузы прямоугольного треугольника по его катетам.)



12. Даны два натуральных числа. Выяснить, является ли хоть одно из них палиндромом ("перевертышем"), т.е. таким числом, десятичная запись которого читается одинаково слева направо и справа налево. (Определить процедуру, позволяющую распознавать числа-палиндромы.)

13. Даны натуральное число *n* и целые числа *а1, а2, ..., ап.* Найти количество чисел *аi* (i = 1, 2, ..., *n),* являющихся степенями пятерки. (Определить процедуру и функцию, позволяющую распознавать степени пятерки.)

14. Найти наибольший общий делитель трех натуральных чисел, имея в виду, что НОД(*а*, *b, с) =* НОД (НОД(*а, b), с*). (Определить функцию и процедуру для расчета наибольшего общего делителя двух натуральных чисел, используя алгоритм Евклида.)

15. Даны основания и высоты двух треугольников. Найти сумму их их площадей. (Определить функцию и процедуру для расчета площади треугольника по его основанию и высоте.)

Лабораторная работа 14 Разработка пользовательских форм

*Цель работы:* научиться создавать пользовательские формы, создавать и использовать элементы управления, ознакомиться с их свойствами и методами

Окно проекта в редакторе VBA активизируется выбором команды **View - Project Explorer**.

В окне проекта представлена иерархическая структура файлов, форм и модулей текущего проекта. В проекте автоматически создается модуль для каждого рабочего листа и для всей книги. Кроме того, модули создаются для каждой пользовательской формы макросов и классов.

Для создания диалоговых окон, разрабатываемых приложений в VBA, используются формы. Редактор форм является одним из основных инструментов визуального программирования. Форма в проект добавляется с помощью команды **Insert - UserForm.** В результате на экран выводится незаполненная форма с панелью инструментов **Toolbox** (рисунок 14.1).

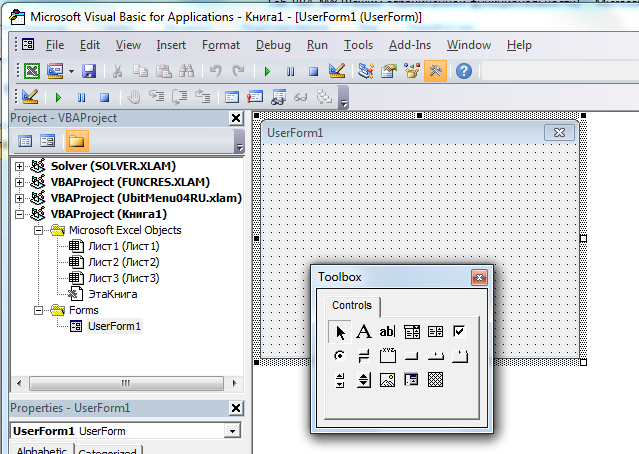


Рисунок 14.1 – Окно конструирования пользовательской формы

Используя панель инструментовиз незаполненной формы, можно сконструировать любое требуемое для приложения диалоговое окно, перетащив на форму нужные элементы управления.

В окне свойств перечисляются основные установки свойств выбранной формы или элемента управления. Используя это окно, можно просматривать свойства и изменять их установки. Окно свойств активируется командой **View – Property Window**.

**Задание**

1. Выполнить задание лабораторных работ №1 и №2 с использованием ввода и вывода на пользовательскую форму.

Форма должна содержать элементы управления:

- TextBox – текстовые поля для ввода и вывода данных

- Label –надписи

- CommandButton – кнопки для выполнения вычислений и очистки текстовых полей

- Image – рисунок с условием задачи

2. Выполнить проверку на ввод нечисловых значений в текстовые поля с использованием функции IsNumeric.

Лабораторная работа 15 Разработка программ для работы с текстовыми данными

*Цель работы*: Освоение основных приемов работы с текстовыми данными и файлами.

VBA позволяет обрабатывать файлы, работать с дисками и каталогами посредством объектной файловой системы (FSO - File System Object  
Model);

FSO предоставляет разработчику возможность работы с файлами в ясной  
объектно-ориентированной манере. FSO опирается на библиотеку Microsoft  
Scripting Runtime Library (файл Scrrun.dll), которая совместно используется  
несколькими средами разработки и позволяет, в частности, расширить  
функции написания сценариев. Для использования этой библиотеки надо  
установить ссылку на **Microsoft Scripting Runtime** в диалоговом окне **Refe-  
rences — VBAProject**, которое отображается на экране выбором команды  
**Tools | References**. После этого можно приступить к программированию объектов, функционирование которых обеспечивается данной библиотекой.

Основными объектами FSO для работы с файлами являются **File**, который является элементом семейства Files, и имеет два свойства Count и Item, и **TextStream**, который позволяет работать с файлами.

Для получения доступа к существующему файлу (объекту **File**) надо воспользоваться методом **GetFile** объекта **FileSystemObject**.

Синтаксис: **GetFile** (filespec)

Прежде чем получать доступ к каталогу, желательно убедиться, что он существует. Это можно сделать при помощи метода **FileExists** объекта **FileSystemObject**.

Синтаксис:

**FileExists**(folderspec)

Метод **FileExists** возвращает True, если диск существует, и False — в противном случае.

Если текстовый файл не существует, то его можно создать при помощи метода **CreateTextFile** объекта **FileSystemObject**. Этот метод создает объект **TextStream** и специфицирует имя создаваемого текстового файла. Объект **Textstream** используется для чтения и записи данных в файл.

Синтаксис: **CreateTextFile**(filename [, overwrite [, unicode]])

filename - строковое выражение, идентифицирующее создаваемый текстовый файл;

overwrite — параметр, принимающий значение равное True, если указанный файл уже существует и его следует перезаписать и False, если существующий файл не перезаписывается.

unicode — определяет код создаваемого текстового файла Если значение параметра равно True, то файл создается как Unicode-файл Если значение параметра равно False, то файл создается как ASCII-файл.

После получения доступа к файлу возможен сбор информации о файле, работа с ним (копирование, удаление, перемещение), а также запись и считывание данных из файла при помощи следующих свойств и методов объекта **File.**

**Delete** - удаляет файл, **Сору** - копирует файл, **Move** - перемещает файл,

**OpenAsTеxtStream** - создает объект **Textstream** и специфицирует имя открываемого текстового файла Этот объект используется для чтения, записи и добавления данных в файл:

**OpenAsTеxtStream** ([iomode, [format]])

• iomode — задает тип операции, производимой над файлом. Допустимые значения:

° ForReadmg или 1 — для чтения,

° ForWntmg или 2 — для записи,

° ForAppending или 8 — для добавления данных.

Метод **OpenTextFile** создает объект **Textstream** и специфицирует имя открываемого текстового файла Этот объект используется для чтения и добавления данных в файл.

*Синтаксис*: **OpenTextFile**(filename[, iomode[, create[, format]]]))

• filename — строковое выражение, идентифицирующее открываемый текстовый файл,

• iomode — задает тип операции, производимой над файлом/ Допустимые значения

° ForReadmg или 1 — для чтения,

° ForAppendmg или 8 — для добавления данных

• create — логический параметр, который указывает, надо ли создавать новый файл, если файл с данным именем не существует,

• format — задает тип файла. Допустимые значения:

° TnstateUseDefault или -2— тип файла, используемыйв системе по умолчанию,

° TnstateTrue или -1 — Unicode-файл,

° TnstateFalse или 0 — ASCII-файл

Для чтения, записи и добавления данных в текстовый файл необходимо создать объект Textstream, ассоциированный с этим файлом, указать совершаемую над ним операцию (запись, добавление данных, чтение), а потом при  
помощи методов и свойств этого объекта реализовать требуемые операции.

**Метод** **Read**(characters) считывает указанное число символов из файла и возвращает результирующую строку.

**Метод** **ReadLine** считывает целую строку из файла и возвращает результирующую строку.

**Метод** **ReadAll** считывает все содержимое файла и возвращает результирующую строку Используется для небольших файлов.

**Метод** **Skip** (characters)- пропускает указанное число символов при считывании данных из файла.

**Метод** **SkipLme** - пропускает одну строку при считывании данных из файла.

**Метод** **Write** (string) - записывает в файл указанную строку, причем в конце строки размещается символ перехода на новую строку

**Метод** **Close -** закрывает открытый Textstream По завершении работы с файлом никогда не забывайте его закрывать

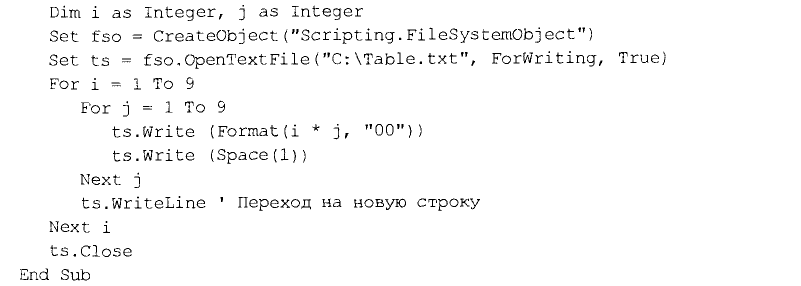
**Свойство** **Line**, **Column** - возвращают номер строки и столбца, в котором находится текущий символ

**Свойство AtEndOfStream** - возвращает True, если указатель достиг конца файла.

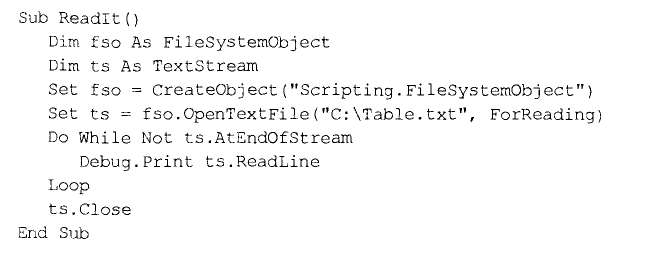
**Свойство AtEndOfLme** - возвращает True, если указатель достиг конца строки

Пример записи в текстовый файл Table.txt создаваемой в программе таблицы умножения представлен ниже.





После того как текстовый файл Table.txt с таблицей умножения создан, построчно вывести его содержимое в окно **Immediate** можно, например, следующим кодом:



**Задание**

1. Создать и выполнить программы из методических указаний.

2. Создать пользовательскую форму с 2 списками и 3 кнопками.

Первый список должен заполняться при инициализации формы из текстового файла. При нажатии на кнопку 1 выбранные элементы из списка 1должны переместиться в список 2 , при нажатии на кнопку 2 выбранные элементы из списка 2 должны переместиться в список 1. При нажатии на кнопку 3 список 2 должен сохраниться в другом текстовом файле. Предусмотреть проверку пустоты списка 2. В случае если он пуст, кнопка 3 должна быть заблокирована для пользователя.

## Список литературы

1. Бычков М. И. Основы программирования на VBA для Microsoft Excel/БычковМ.И. - Новосиб.: НГТУ, 2010. - 99 с.
2. Назаров, С. В. Программирование в пакетах MS Office [Электронный ресурс] : учеб. пособие / С. В. Назаров, П. П. Мельников, Л. П. Смольников и др.; под ред. С. В. Назарова. - М.: Финансы и статистика, 2007. - 656 c.: ил. - ISBN 978-5-279-02926-6
3. Информатика: Учебник / В.А. Каймин; Министерство образования РФ. - 6-e изд. - М.: ИНФРА-М, 2010. - 285 с.
4. Excel 2010 на примерах: Практическое пособие / Васильев А.Н. - СПб:БХВ-Петербург, 2010. - 422 с.
5. Информатика (курс лекций): учебное пособие / В.Т. Безручко. - М.: ИД ФОРУМ: ИНФРА-М, 2012. - 432 с.: ил.