МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ИНФОРМАТИКА

Методические рекомендации к лабораторным работам для студентов специальности 6-05-0714-02 «Технология машиностроения, металлорежущие станки и инструменты» очной формы обучения



Могилев 2024

УДК 004 ББК 32.973 И74

Рекомендовано к изданию учебно-методическим отделом Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «26» апреля 2024 г., протокол № 10

Составитель ст. преподаватель В. М. Прудников

Рецензент канд. техн. наук, доц. И. В. Лесковец

Методические рекомендации к лабораторным работам по дисциплине «Информатика» предназначены для студентов специальности 6-05-0714-02 «Технология машиностроения, металлорежущие станки и инструменты» очной формы обучения. Приведен список литературы для подготовки к лабораторным работам.

Учебное издание

ИНФОРМАТИКА

Ответственный за выпуск	В. В. Кутузов
Корректор	И. В. Голубцова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс. . Уч.-изд. л. . Тираж 16 экз. Заказ №

Издатель и полиграфическое исполнение: Межгосударственное образовательное учреждение высшего образования «Белорусско-Российский университет». Свидетельство о государственной регистрации издателя, изготовителя, распространителя печатных изданий № 1/156 от 07.03.2019. Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский университет, 2024

Содержание

Введение	4
1 Лабораторная работа № 1. Текстовый процессор MS Word.	
Элементы окна	5
2 Лабораторная работа № 2. Текстовый процессор MS Word.	
Работа с таблицами	7
3 Лабораторная работа № 3. Текстовый процессор MS Word.	
Создание базы данных. Слияние документов	8
4 Лабораторная работа № 4. Текстовый процессор MS Word.	
Работа с графическими объектами и гипертекстом	9
5 Лабораторная работа № 5. Табличный процессор MS Excel.	
Элементы окна. Ввод и редактирование данных и формул	11
6 Лабораторная работа № 6. Табличный процессор MS Excel.	
Построение диаграмм	13
7 Лабораторная работа № 7. Табличный процессор MS Excel.	
Создание баз данных. Сортировка и фильтрация данных	14
8 Лабораторная работа № 8. Табличный процессор MS Excel.	
Использование надстроек	16
9 Лабораторная работа № 9. Алгоритмический язык VBA.	
Ознакомление с интегрированной системой программирования	22
10 Лабораторная работа № 10. Алгоритмический язык VBA.	
Разработка разветвленной программы с использованием оператора IF	25
11 Лабораторная работа № 11. Алгоритмический язык VBA.	
Разработка разветвленной программы с использованием оператора	
выбора	26
12 Лабораторная работа № 12. Алгоритмический язык VBA.	
Разработка программы с использованием оператора выбора FOR. Ввод	
и вывод одномерного массива	28
13 Лабораторная работа № 13. Алгоритмический язык VBA.	
Обработка одномерного массива	30
14 Лабораторная работа № 14. Алгоритмический язык VBA.	
Разработка программы с вложенными циклами. Обработка многомерных	
массивов	33
15 Лабораторная работа № 15. Алгоритмический язык VBA.	
Разработка программы с использованием циклов While	34
16 Лабораторная работа № 16. Работа в MathCad. Использование	
пакета Mathcad для решения математических задач	37
Список литературы	39

Введение

Цель методических рекомендаций – помочь студентам в подготовке к выполнению лабораторных работ по дисциплине «Информатика».

Цель преподавания дисциплины – изучение основных современных операционных систем и программных сред, пакетов прикладных программ для научных и инженерных расчетов, основ программирования, общих вопросов алгоритмизации и приобретение навыков решения задач с применением средств вычислительной техники.

Дисциплина «Информатика» является неотъемлемой частью современных инженерных знаний и входит в состав естественно-научных дисциплин, компонентов учреждения высшего образования.

Полученные при изучении дисциплины знания и навыки востребованы при изучении специальных дисциплин инженерной направленности и станут инструментом для грамотного выполнения и оформления рефератов, курсовых и дипломных работ.

1 Лабораторная работа № 1. Текстовый процессор MS Word. Элементы окна

Цель работы: изучение элементов окна MS Word и приобретение навыков работы с текстовым процессором MS Word.

Порядок выполнения работы

1 Ознакомиться с основными теоретическими положениями.

2 Выполнить задание, полученное у преподавателя.

3 Оформить отчет в виде набора файлов.

Основные теоретические положения

1 Элементы окна MS Word.

Строка заголовка расположена в верху окна. В этой строке указываются название программы (Microsoft Word) и имя обрабатываемого файла.

Строка основного меню расположена под строкой заголовка. В основном меню перечислены команды работы с текстом.

Пиктографическое меню (панели инструментов) расположено ниже строки основного меню и содержит кнопки-пиктограммы. После щелчка мыши по пиктограмме выполняется определенная команда редактора MS Word. Пиктографическое меню является альтернативой строке основного меню редактора и предназначено для ускоренного запуска команд.

Редактор Word предоставляет пользователю возможность самому компоновать пиктографическое меню. В случае отсутствия требуемой панели на экране, ее можно установить командой *Bud – Панели инструментов*.

Обычно при работе с текстом используются панели *Стандартная* и Форматирование. Панель *Рисование* активизируется для рисования графических объектов.

Линейки, как правило, расположены сверху и слева от редактируемого документа. С помощью линеек изменяются абзацный отступ, ширина колонок текста и размеры ячеек таблиц, а также устанавливаются позиции табуляции в тексте. Для изменения абзацных отступов и границ текста достаточно подвести курсор мыши к указателям и, удерживая нажатой левую кнопку, переместить их в нужную позицию.

Линейки устанавливаются с помощью команды *Вид – Линейки*.

Полосы прокрутки. Справа и внизу от документа расположены полосы вертикальной и горизонтальной прокрутки текста. Они применяются в случае, если текст целиком не помещается на экране. Для прокрутки текста используется бегунок, передвигающийся по полосе. Перетаскивая мышью бегунок, пользователь имеет возможность быстро перемещаться по тексту документа.

Строка состояния расположена в нижней части окна редактора Word. В эту строку выводится информация о текущем положении текстового курсора в документе, текущем времени и режиме редактирования.

Текстовый курсор представляет собой вертикальный штрих. Его место-

положение определяет позицию, с которой будет записываться информация, поступающая с клавиатуры, включаться рисунок или таблица. Курсор перемещается с помощью клавиш управления курсором или мыши.

В пунктах Основного меню редактора Word перечислены все команды редактора.

2 Порядок работы в текстовом редакторе Word.

2.1 Создать документ. Новый документ создается командой меню Файл – *Создать*. В результате на экране появляется диалоговое окно *Создать*, в котором пользователю требуется выбрать шаблон документа (например, normal) и нажать кнопку *ОК*.

2.2 Определить параметры страницы документа. Перед набором текстового документа установить параметры страницы документа.

Для этого надо использовать команду *Файл – Параметры страницы*. В результате появится диалоговое окно с вкладками: *Поля*, *Размер бумаги*, Источник бумаги, Макет.

Используя вкладку Поля, пользователь задает границы области текста в документе. С помощью вкладки Размер бумаги определяются размер бумажного листа и его ориентация. Редактор Word позволяет создавать документы на листах формата A5 (размер соответствует почтовой карточке), формата A4 (обычный печатный лист), формата A3 и т. д., или установить собственный размер листа.

2.3 Установить шрифт и его параметры (шрифт, начертание, размер, цвет, подчеркивание, эффекты). Для этого используется команда Формат – Шрифт. Если необходимо переустановить параметры шрифта в набранном фрагменте текста, то предварительно этот фрагмент выделяется.

2.4 Определить формат абзаца (способ выравнивания текста, отступы и интервалы). Для достижения этой цели необходимоу становить курсор ввода в нужный абзац и выполнить команду *Формат – Абзац*. В диалоговом окне установить требуемые значения в полях *Выравнивание*, *Отступ*, *Интервал*.

2.5 Набрать текст документа.

2.6 Проверить правописание в документе командой Сервис – Правописание.

2.7 Сохранить документ. Сохранение осуществляется выбором команды Файл – Сохранить (сохранить под старым именем) или Файл – Сохранить как ... (сохранить с новыми параметрами – указываются новые папка, имя, тип и т. д.).

В результате на экране появляется диалоговое окно *Сохранение документа*. Это окно содержит поля, в которых задается имя файла и маршрут к нему (его адрес), указывается тип сохраняемого файла.

2.8 Подготовить документ для вывода на печать. Перед началом печати документа, рекомендуется проверить правильность расположения на странице рисунков, абзацев и т. п. Для этого следует использовать режим предварительного просмотра. Чтобы переключить в этот режим, требуется выбрать команду Файл – Просмотр. Такую проверку можно выполнить при одновременном изображении на экране нескольких страниц документа.

Печать документа осуществляется с использованием команды основного меню *Файл – Печать*. После выбора команды появляется диалоговое окно *Печать*.

Контрольные вопросы

- 1 Какие элементы окна MS Word используются?
- 2 Каков порядок работы в текстовом процессоре MS Word?
- 3 Какие варианты для создания документа?

2 Лабораторная работа № 2. Текстовый процессор MS Word. Работа с таблицами

Цель работы: приобретение практических навыков по работе с таблицами в процессоре Microsoft Word.

Порядок выполнения работы

1 Ознакомиться с основными теоретическими положениями.

2 Выполнить задание, полученное у преподавателя.

3 Оформить отчет в виде набора файлов.

Основные теоретические положения

Таблицы предназначены для упорядочивания данных и создания интересных макетов страницы с последовательно расположенными столбцами текста или графики. Для работы с таблицами используется пункт меню *Табли*иа. Наиболее быстрый путь создания простой таблицы – например такой, которая имеет одинаковое количество строк и столбцов – с помощью команды *Добавить таблицу*.

Создание простой таблицы:

- выберите место создания таблицы;

- выберите команду Добавить таблицу из меню Таблица;

– установите нужные параметры (число столбцов и строк, ширина столбцов, автоформат таблицы).

Добавление строк и столбцов в таблицу:

– чтобы добавить строку в конец таблицы, щелкните последнюю ячейку последней строки, а затем нажмите *Tab*;

– чтобы добавить столбец справа от последнего столбца в таблице, щелкните за пределами самого правого столбца. Выберите команду Выделить столбец в меню Таблица, а затем нажмите команду Добавить столбцы.

Изменение ширины столбца или высоты строки таблицы:

– чтобы для нескольких столбцов (строк) установить одинаковую ширину (высоту), выделите нужные столбцы (строки), а затем выберите команду Выровнять ширину столбцов (Выровнять высоту строк) в меню Таблица;

– можно также изменить ширину столбцов (высоту строк) при помощи перетаскивания границ столбца (строки) в самой таблице или перетаскиванием маркеров границ столбцов (строк) таблицы на горизонтальной (вертикальной) линейке.

Отображение или скрытие сетки в таблице: выберите команду *Отобра*жать сетку или Скрыть сетку в меню Таблица.

Удаление из таблицы ячеек, строк или столбцов:

 выделите ячейки, строки или столбцы для удаления. При удалении ячеек включите режим отображения символов ячейки. При удалении строк включите режим отображения символов строк;

– выберите команду Удалить ячейки, Удалить строки или Удалить столбцы в меню Таблица;

– при удалении ячеек выберите нужный параметр (со сдвигом влево или вправо, удалить всю строку или весь столбец).

Контрольные вопросы

1 Какой пункт меню Word используется для работы с таблицами?

2 Какие клавиши используются для перемещения в следующую и предыдущую ячейки таблицы?

3 Как добиться, чтобы все ячейки были одинаковой ширины?

4 Как удалить строку или столбец из таблицы?

5 Как объединить несколько ячеек таблицы?

3 Лабораторная работа № 3. Текстовый процессор MS Word. Создание базы данных. Слияние документов

Цель работы: приобретение практических навыков создания баз данных и слияния документов.

Порядок выполнения работы

1 Ознакомиться с основными теоретическими положениями.

2 Выполнить задание, полученное у преподавателя.

3 Оформить отчет в виде набора файлов.

Основные теоретические положения

MS WORD позволяет создавать простейшее базы данных и объединять документ, содержащий неизменную информацию (например, бланк платежного поручения, бланк задания на курсовое проектирование) и информацию из базы данных (например, фамилию студента, задачу для курсового проектирования и т. д.). Таким образом можно печатать большое количество однотипных документов.

Процесс объединения бланка и информации из базы данных называется слиянием.

Подготовка документов путем *слияния* применяется в тех случаях, когда один и тот же документ требуется разослать большому числу адресатов. В такой ситуации поступают следующим образом.

Готовят так называемый *основной документ*, который содержит общую для всех адресатов информацию.

Затем формируют *источник данных*, который содержит настраиваемую на адресата информацию, как правило, это адрес получателя и некоторые другие реквизиты.

На заключительном этапе результирующий документ для каждого адресата формируется автоматически на основе слияния основного документа и данных из источника данных, относящихся именно к этому адресату.

Источник данных содержит таблицу записей с текстом и/или графикой, каждая из которых попадает в отдельный документ слияния.

Такая технология обычно применяется для создания конвертов, наклеек и документов на бланках.

В редакторе MS Word для этих целей используется лента *Рассылки*, включающая группы *Создать*, *Начать слияние*, *Составление документа и вставка полей*, *Просмотр результатов* и *Завершить*.

Подготовка составного документа включает следующие этапы.

1 Создание основного документа.

2 Задание источника данных.

3 Подготовка основного документа к слиянию.

4 Слияние основного документа и данных из присоединенного источника данных.

Контрольные вопросы

1 Для чего используется слияние документов?

2 Что такое база банных (источник данных)?

3 Что такое основной документ?

4 Каковы этапы подготовки составного документа?

4 Лабораторная работа № 4. Текстовый процессор MS Word. Работа с графическими объектами и гипертекстом

Цель работы: изучение возможностей работы с графическими объектами и гипертекстовыми документами.

Порядок выполнения работы

1 Ознакомиться с основными теоретическими положениями.

2 Выполнить задание, полученное у преподавателя.

3 Оформить отчет в виде набора файлов.

Основные теоретические положения

Наличие графических объектов в текстовых документах часто желательно, а в некоторых случаях просто необходимо. Текстовый редактор MS Word предоставляет достаточно большие возможности работы с объектами как растровой графики (построенными с помощью отдельных точек – пикселей), так и векторной (построенными на основе графических примитивов).

Основные инструменты для работы с графикой находятся на вкладке Вставка в группе Иллюстрации, содержащей элементы Рисунок, Клип, Фигуры, SmartArt и Диаграмма.

Для вставки рисунка из файла можно воспользоваться меню Вставка – Иллюстрации – Рисунок. После вставки рисунка можно установить для него формат: щелкнуть правой кнопкой мыши по вставленному рисунку, выбрать команду Формат рисунка, установить необходимые параметры и нажать кнопку Закрыть. Можно также определить способ обтекания текстом вставленного рисунка, выбрав команду Обтекание текстом из контекстного меню.

Удобства работы с текстовыми документами, такими как электронные учебники, мультимедийные справочники и т. д., в редакторе MS Word обеспечиваются системой навигации с использованием гиперссылок. Гиперссылки могут быть построены на словах текста, фразах или рисунках, которые содержат внедренные в них адреса URL и позволяют организовать переход к любой закладке текста, файлу или ресурсу сети Интернет.

Чтобы построить гиперссылку на раздел текущего документа, сначала необходимо создать закладку на каком-либо его слове или фразе. Для этого следует выделить это слово или фразу и на вкладке *Вставка* в группе *Связи* кликнуть на элементе Закладка, ввести имя закладки и щелкнуть по кнопке Добавить.

После этого можно построить гиперссылку: выделить текстовый элемент (слово или фразу), кликнуть на нем правой кнопкой мыши, выбрать *Гиперссылка*, затем в открывшемся окне Вставка гиперссылки щелкнуть по кнопке Закладка или выбрать команду Связать с – Места в документе, кликнуть нужную закладку и кнопку OK.

В окне *Вставка гиперссылки* можно построить гиперссылку на файл или web-страницу, новый документ или электронную почту.

Контрольные вопросы

1 Назовите основные элементы группы Иллюстрации вкладки Вставка.

2 Какие группы графических элементов содержаться в меню Фигуры?

- 3 Как нарисовать эллипс и залить его синим цветом?
- 4 Как изменить толщину уже существующей линии?
- 5 Что понимается под гиперссылкой?

5 Лабораторная работа № 5. Табличный процессор MS Excel. Элементы окна. Ввод и редактирование данных и формул

Цель работы: изучение основных понятий и элементов окна MS Excel, а также порядка ввода, форматирования, редактирования данных и работы с формулами в MS Excel.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить отчет в виде набора файлов.

Основные теоретические положения

Документ Excel называется *рабочей книгой*. Каждая рабочая книга состоит из одного или нескольких *рабочих листов*. По умолчанию таких листов три (с именами Лист1 – Лист3) и все они пусты. Манипуляции с рабочими листами производятся с помощью *ярлычков* рабочих листов, расположенных внизу окна рабочей книги.

Рабочий лист – это множество элементарных ячеек, каждая из которых принадлежит некоторому столбцу и одновременно принадлежит некоторой строке.

Все строки содержат одинаковое количество ячеек и все столбцы содержат одинаковое количество ячеек, т. е. рабочий лист имеет прямоугольную форму. По умолчанию рабочий лист имеет 256 столбцов с именами от А до IV и 65536 строк с номерами от 1 до 65536 (этого достаточно в большинстве случаев). Если распечатать всю таблицу, понадобится лист шириной 6,4 м и длиной 104 м. На экране монитора видно ≈ 4 % ширины и 0,1 % длины таблицы.

Столбцы именуются буквами, а строки нумеруются цифрами. Имя столбца и номер строки в совокупности однозначно идентифицируют ячейку, которая им (одновременно) принадлежит. Этот идентификатор называется *адресом* ячейки или *ссылкой* на ячейку. Например, ячейка, находящаяся в четвёртой строке в столбце С, имеет адрес С4. Такой стиль ссылок в Excel называется A1. Адрес (имя) ячейки отображается в левой части строки формул.

Кроме понятия ячейки, используется понятие *диапазона* (или *интервала*) ячеек – прямоугольной области смежных ячеек. Диапазон задается указанием адреса верхней левой ячейки и правой нижней ячейки, разделенных символом «:» (двоеточие).

1 Основное меню.

Каждый пункт системы меню имеет свое подменю команд, которые позволяют выполнять различные действия в Excel.

Меню Файл включает команды, предназначенные для работы с файлами.

Меню Правка объединяет команды для использования буфера обмена и корректировки таблицы.

Меню *Вид* определяет интерфейс пользователя (внешний вид окна Excel).

Меню *Вставка* (позволяет вставить в таблицу ячейки, строки, столбцы, листы, диаграммы, макросы, разрыв страницы, функции, определять имена ячеек и диапазонов).

Меню Формат (позволяет определить формат ячеек строк, столбцов, листов).

Меню *Окно* осуществляет управление окнами: изменение их размеров, расположения и т. д., упорядочивание расположения окон, разделение окна на части и др.

2 Ввод данных.

Для ввода данных в ячейки рабочего листа нужно поступить следующим образом.

Вначале следует ячейку, в которую вводятся данные, сделать активной (выделить), для чего следует переместить указатель мыши, имеющий вид жирного креста ⁽¹⁾, в эту ячейку и один раз щелкнуть левой кнопкой мыши.

После этого можно либо прямо вводить данные с клавиатуры (при этом курсор ввода находится внутри ячейки – такой режим называется *редактированием в ячейке*), либо щелкнуть левой кнопкой мыши в строке формул и после этого вводить данные (при этом курсор ввода находится в *строке формул* над рабочим листом – такой режим называется *редактированием в строке формул*).

3 Ввод и редактирование формул.

Формулой в Excel называется последовательность символов, начинающихся со знака равенства «=». В эту последовательность символов могут входить постоянные значения (константы), ссылки на ячейки, имена, функции или операторы.

Результатом работы формулы является новое значение, которое выводится как результат вычисления формулы по уже имеющимся данным.

4 Перемещение, копирование и распространение формул.

После того как формула введена в ячейку, можно ее *перенести*, *скопировать* или *распространить* на блок ячеек.

При *перемещении* формулы в новое место таблицы ссылки в формуле не изменяются, а ячейка, где раньше была формула, становится свободной.

При *копировании* формула перемещается в другое место таблицы, ссылки изменяются, но ячейка, где раньше находилась формула, остается без изменений. Формулу можно *распространить* (размножить) на блок ячеек.

Контрольные вопросы

1 Перечислите состав элементов окна MS Excel.

- 2 Что такое абсолютный и относительный адрес ячеек таблицы?
- 3 Как ввести/удалить формулу?

4 Какие категории встроенных функций программы MS Excel Вы знаете?

6 Лабораторная работа № 6. Табличный процессор MS Excel. Построение диаграмм

Цель работы: изучение порядка построения и форматирования диаграмм в MS Excel.

Порядок выполнения работы

1 Ознакомиться с основными теоретическими положениями.

2 Выполнить задание, полученное у преподавателя.

3 Оформить отчет в виде набора файлов.

Основные теоретические положения

Для построения диаграмм предназначена программа *Мастер диаграмм*, которая осуществляет пошаговое руководство процессом создания диаграммы.

Для достижения желаемых результатов информацию нужно представить в виде сплошной таблицы. Перед тем как вызвать *Мастер диаграмм*, нужно выделить интервал (диапазон) ячеек, информация о которых должна использоваться при создании диаграммы.

Для построения диаграммы выполните следующие действия.

1 Выделите область данных (таблицу), по которым будет строиться диаграмма.

2 Щелкните указателем мыши по кнопке *Мастер диаграмм*. Excel выведет диалоговое окно *Мастер диаграмм* (шаг 1 из четырёх): тип диаграммы.

3 Выберите нужный тип диаграммы. Каждый тип диаграммы имеет несколько различных видов. Для каждого типа диаграммы в нижней части окна дается его краткое описание.

Выберите вид диаграммы. При нажатии на кнопку Просмотр результата можно увидеть как будет выглядеть диаграмма при выбранных типе и виде.

4 Щелкните на кнопке Далее или нажмите Enter.

Появится диалоговое окно *Мастер диаграмм* (шаг 2 из четырёх): источник данных диаграммы.

Пока выведено на экран диалоговое окно *Мастера диаграмм*, проверьте, чтобы интервал ячеек, выделенный на рабочей таблице (пункт 1), соответствовал записи в поле *Диапазон* (с абсолютными ссылками на клетки).

5 Щелкните по кнопке Далее или нажмите Enter. Отобразится диалоговое окно Мастер диаграмм (шаг 3 из четырёх): параметры диаграммы, где можно выполнить добавление заголовков и определить параметры элементов диаграммы – Оси, Линии сетки, Легенда, Подписи данных и Таблицы данных.

Можно ввести заголовок для всей диаграммы так же, как и индивидуальные заголовки для независимой (ось X) и зависимой (ось Y) переменной.

6 Щелкните по кнопке *Далее* или нажмите *Enter*. Появится диалоговое окно *Мастер диаграмм* (шаг 4 из четырёх): размещение диаграммы, где можно определить место размещения диаграммы.

7 Щелкните по кнопке Готово или нажмите Enter, чтобы закрыть диалоговое

окно Мастер диаграмм. Созданная диаграмма появится на рабочем листе.

Можно легко передвигать диаграмму или изменять ее размер сразу после создания, т. к. диаграмма по-прежнему активна. После того как диаграмма приняла нужные размеры и находится в нужном месте, оставьте ее там, сделав неактивной (щелкнув указателем мыши на любую клетку вне диаграммы).

Контрольные вопросы

- 1 Для чего предназначены диаграммы?
- 2 Каков порядок построения диаграммы?
- 3 Как изменить форматирование диаграммы?

7 Лабораторная работа № 7. Табличный процессор MS Excel. Создание баз данных. Сортировка и фильтрация данных

Цель работы: изучение принципов работы с базами данных в MS Excel.

Порядок выполнения работы

1 Ознакомиться с основными теоретическими положениями.

2 Выполнить задание, полученное у преподавателя.

3 Оформить отчет в виде набора файлов.

Основные теоретические положения

1 База данных.

Одна из задач, для которых часто используется Excel, – это организация баз данных.

Термин «база данных» можно применить к любой совокупности связанных данных, объединенных вместе по определенному признаку.

Основным назначением баз данных является обеспечение ввода, хранения и быстрого поиска содержащихся в ней данных.

В Excel термины база данных и список часто используются как синонимы.

В Excel списком называется снабженная метками последовательность строк рабочего листа, содержащих в одинаковых столбцах данные одного типа.

Построение табличных баз данных (списков) начинается с заполнения строки с заголовками столбцов, которые называются *имена полей*. Они обозначают различные типы объектов в том наборе данных, с которыми Вы хотите работать.

После ввода строки с именами полей можно начать ввод информации в базу данных, размещая её под соответствующими именами полей. Функцию *автозавершения* целесообразно применять при формировании списка, который содержит повторяющиеся значения.

Каждый столбец в базе данных содержит информацию определенного типа. Столбец в базах данных известен под именем *поле*. Данные, записанные в строках, называют *записями*. Поле – столбец, содержащий данные определенного типа.

Имя поля – заголовок столбца.

Записи – данные, записанные в строках.

2 Поиск записей.

Для поиска записей с помощью формы данных используются критерии поиска.

В качестве критерия поиска используется либо известная информация, либо символы «*» и «?».

Для сужения области поиска можно указать в различных полях несколько критериев.

3 Сортировка базы данных.

Каждая база данных имеет некоторый порядок поддержания и просмотра записей. После ввода данных Вам может понадобиться упорядочить их. Процесс упорядочивания записей называется сортировкой. Но Вы должны иметь возможность восстановить исходный порядок записей. Универсальным средством для этого является введение порядковых номеров записей.

Сортировка устанавливает порядок строк в таблице в соответствии с содержимым конкретных столбцов, но можно отсортировать и столбцы.

Тип данных влияет на результат сортировки. Для получения правильного результата все ячейки в столбце должны содержать один и тот же тип данных.

При выборе команды *Данные – Сортировка* открывается окно диалога «Сортировка диапазона», в котором надо указать поля сортировки и определить критерий сортировки.

С помощью раскрывающегося списка *Сортировать по* можно выбрать столбец для сортировки. Порядок сортировки устанавливается переключателями *По возрастанию* или *По убыванию*.

Сортировка по возрастанию предполагает следующий порядок:

– числа;

- текст (включая текст с числами);

– логические значения;

– значения ошибок;

– пустые ячейки.

При сортировке по возрастанию текстовые данные упорядочиваются в алфавитном порядке от А до Я. Числовые данные упорядочиваются по возрастанию значения от минимального до максимального.

При выборе переключателя *По убыванию* порядок сортировки меняется на противоположный. Исключением являются пустые ячейки, которые всегда располагаются в конце списка.

Два дополнительных раздела Затем и В последнюю очередь позволяют определить порядок вторичной сортировки записей, в которой имеются совпадающие значения.

Окно диалога *Сортировка диапазона* содержит кнопку *Параметры*, в результате нажатия которой открывается окно диалога *Параметры сортировки*. С помощью этого окна можно:

– определить пользовательский порядок сортировки для столбца, указан-

ного в раскрывающемся списке Сортировать по;

– сделать сортировку чувствительной к использованию прописных и строчных букв;

– изменить направление сортировки (вместо сортировки сверху вниз установить сортировку слева направо).

4 Фильтрация данных в списке.

Фильтрация списка позволяет находить и отбирать для обработки часть записей в списке (базе данных). В отфильтрованном списке выводятся на экран только те строки, которые содержат определенное значение или отвечают определенным критериям. При этом остальные строки оказываются скрытыми.

Все это упрощает процесс ввода и удаления записей, а также процесс поиска информации.

Преимущество применения фильтров состоит в том, что результат фильтрации можно скопировать в отдельную область таблицы и сразу же использовать в вычислениях.

В Excel для фильтрации данных используются команды Автофильтр и Расширенный фильтр. В случае простых критериев для выборки нужной информации достаточно команды Автофильтр. При использовании сложных критериев следует применять команду Расширенный фильтр.

Обе команды вызываются в результате выбора команды Данные – Фильтр.

Контрольные вопросы

1 Как создать форму данных для новой базы данных?

- 2 Для чего используется сортировка базы данных?
- 3 Какие команды используются для фильтрации данных?
- 4 В каких случаях применяется команда Расширенный фильтр?

8 Лабораторная работа № 8. Табличный процессор MS Excel. Использование надстроек

Цель работы: приобретение навыков использования надстроек для решения уравнений и систем уравнений с помощью подбора параметра и поиска решения.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить отчет в виде набора файлов.

Основные теоретические положения

Надстройки – это специальные средства, расширяющие возможности программы Excel. На практике именно надстройки делают программу Excel

удобной для использования в научно-технической работе.

Специальная функция Подбор параметра позволяет определить параметр (аргумент) функции, если известно ее значение. При подборе параметра значение влияющей ячейки (параметра) изменяется до тех пор, пока формула, зависящая от этой ячейки, не возвратит заданное значение. Другими словами, данный инструмент следует применять для анализа данных с одним неизвестным (или изменяемым) условием.

Технология использования команды следующая.

1 Выделить ячейку с формулой, которая должна принять заданное значение (целевую ячейку).

2 Выбрать команду с вкладки Данные \rightarrow Работа с данными \rightarrow Анализ «что-если» \rightarrow Подбор параметра. В появившемся после этого диалоговом окне в поле Установить в ячейке уже будет находиться ссылка на выделенную при выполнении шага 1 ячейку.

3 В поле *Значение* ввести величину, которую необходимо получить в целевой ячейке.

4 В поле *Изменяя значение ячейки* ввести ссылку на ячейку-параметр (данная ячейка не должна содержать формулу).

Пример – Решить уравнение 2x + 1 = 7.

y = 7 является функцией x, т. е. известно значение y. Следует узнать, при каком значении x мы получим y, вычисляемый формулой.

Решим данную задачу встроенными вычислительными инструментами Excel для анализа данных Подбор параметра.

1 Запустите программу Excel и откройте рабочую книгу. Присвойте рабочему листу имя *Уравнение*.

2 Заполните ячейки листа так, как показано на рисунке 1.

0	f_{x}	=2*A2+1
	А	В
1	х	формула
2		1

Рисунок 1 – Подготовка к подбору параметра

3 Перейдите в ячейку В2 и выберите инструмент, где находится подбор параметра в Excel: Данные \rightarrow Работа с данными \rightarrow Анализ «что-если» \rightarrow Подбор параметра.

4 В появившемся диалоговом окне *Подбор параметра* заполните поля значениями, как показано на рисунке 2, и нажмите *OK*.

5 В результате мы получим правильное решение, которое будет записано в ячейку А2, как показано на рисунке 3.

Excel в своих алгоритмах инструментов анализа данных использует простой метод – подстановки. Он подставляет вместо *x* разные значения и анализирует, насколько результат вычислений отклоняется от условий, указан-

ных в параметрах инструмента. Как только будет достигнут результат вычисления с максимальной точностью, процесс подстановки прекращается.

Данные Рецензирован	ние Вид	Разрабо	тчик	Office Tab	Hag
Очистить			👸 Прове	рка данных 👻	4
Повторить			🚰 Консол	пидация	4
Дополнительно	Текст по 3 столбцам ду	/далить бликаты 🗄	🖗 Анали	з "что если" 👻	1000 1000 1000 1000
вка и фильтр		Работа	Дис	петчер сценар	иев.
Подбор параметра		? ×	Под	бор параметр	a
Установить в <u>я</u> чейке:	B2	<u>B</u>			
Зна <u>ч</u> ение:	7				
Изменяя значение ячейки:	\$A\$2	B			

Рисунок 2 – Диалоговое окно Подбор параметра



Рисунок 3 – Результат подбора параметра

По умолчанию инструмент выполняет 100 повторений (итераций) с точностью 0,001. Если нужно увеличить количество повторений или повысить точность вычисления измените настройки: $\Phi a \ddot{u} \rightarrow \Pi a p a mempia \rightarrow \Phi o p mynia \rightarrow \Pi a p a mempia вычислений.$

1 Увеличить в настройках параметр предельного числа итераций.

2 Изменить относительную погрешность.

3 В ячейке переменной (как во втором примере, А3) ввести приблизительное значение для быстрого поиска решения. Если же ячейка будет пуста, то Excel начнет с любого числа.

Специальная функция *Поиск решений* может применяться для решения задач, которые включают много изменяемых ячеек, и помогает найти комбинацию переменных, которые максимизируют или минимизируют значение в целевой ячейке. Он также позволяет создать одно или несколько ограничений – условий, которые должны выполняться при поиске решений.

Подключение осуществляется через вкладку $\Phi a \ddot{u} n \to \Pi a pamempu \to Had$ $стройки. В открывшемся окне выберите необходимые надстройки <math>\Pi ouck$ решения и нажмите кнопку $\Pi e pe \ddot{u} m u$. В появившемся окне отметьте галочками нужный надстройки и нажмите OK. Выбранные вами надстройки отобразятся на вкладке $\Pi a m u$.

Технология использования команды Поиск решения следующая.

1 Выделить ячейку с формулой, которая должна принять заданное значение (целевую ячейку).

2 Выбрать команду меню Данные → Анализ → Поиск Решения. В появившемся после этого диалоговом окне в поле Установить в ячейке уже будет находиться ссылка на выделенную при выполнении шага 1 ячейку.

3 В поле Изменяя ячейки следует задать ячейки с переменными или нажать кнопку Предположить, и поиск решения сам предложит изменяемые ячейки, исходя из заданной целевой функции. Поле Изменяя ячейки нельзя оставлять пустым, и указанные в нем ячейки обязательно должны влиять на значение целевой ячейки.

4 Последний шаг определения поиска решений – задание ограничений. Он не является обязательным. Чтобы задать ограничения, следует в диалоговом окне Поиск решения нажать кнопку Добавить и заполнить окно диалога Добавление ограничений.

5 После заполнения диалогового окна Поиск Решения следует нажать кнопку Выполнить. При нахождении оптимального решения на экран выводится диалоговое окно Результаты поиска решения. Значения, отображаемые на рабочем листе, представляют собой оптимальное решение задачи. Можно либо оставить эти значения на листе, установив переключатель Сохранить найденное решение и нажав кнопку ОК. Либо восстановить исходные значения, установив переключатель Восстановить исходные значения или нажав кнопку Отмена.

Параметры инструмента Поиск решения.

Как отмечалось ранее, доступ к инструменту Поиск решения осуществляется с помощью команды Данные \rightarrow Анализ \rightarrow Поиск решения. Данная команда отображает диалоговое окно (ДО) Параметры поиска решения.

Перед использованием рассматриваемого инструмента на листе электронной таблицы должны быть сформированы целевая функция, область изменяемых ячеек (неизвестные), значения которых будут найдены в процессе решения. Решение (изменяемые ячейки) должно находиться в определенных пределах или удовлетворять определенным ограничениям.

Параметры задачи ограничиваются следующими предельными показателями: количество неизвестных – 200; количество формульных ограничений на неизвестные – 100; количество предельных условий на неизвестные – 400.

В ДО Параметры поиска решения в поле Оптимизировать целевую функцию указывается адрес ячейки с целевой функцией. Целевая функция зависит от изменяемых ячеек и связана с ними некоторой формулой. Оптимизируется значение целевой функции до максимума, минимума или некоторого определенного значения. В поле Изменяя ячейки переменных указывается адрес блока ячеек, которые и будут решением.

В область В соответствии с ограничениями вводятся ограничения на решение. Кнопки Добавить, Изменить, Удалить управляют ограничениями.

Флажок в поле *Сделать переменные без ограничений неотрицательными* позволяет не вводить дополнительно ограничения на изменяемые ячейки, если их значения неотрицательны.

Поиск решения, в зависимости от типа решаемых задач, позволяет исполь-

зовать методы: симплексный метод, метод ОПГ (обобщенного приведенного градиента), эволюционный поиск решения.

Метод решения выбирается из раскрывающегося списка *Выберите метод решения* рассматриваемого окна диалога. Кнопка *Найти решение* запускает процесс решения задачи.

Сохранить модель поиска решения можно следующими способами:

1) при сохранении книги Excel после поиска решения все значения, введенные в ДО *Поиск решения*, сохраняются вместе с данными рабочего листа. С каждым рабочим листом в рабочей книге можно сохранить один набор значений параметров *Поиска решения*;

2) если в пределах одного рабочего листа Excel необходимо рассмотреть несколько моделей оптимизации (например, найти максимум и минимум одной функции или максимальные значения нескольких функций), то удобнее сохранить эти модели, используя кнопку *Параметры/Сохранить модель окна Поиск решения*. Выбор модели для решения конкретной оптимизационной задачи осуществляется с помощью кнопки *Параметры/Загрузить модель* ДО *Поиск решения*;

3) также можно сохранить данные в виде именованных сценариев. Для этого необходимо нажать на кнопку *Сохранить сценарий* ДО *Результаты поиска решений*.

Иногда в результате выполнения процедуры поиска решения само решение не находится, даже если известно, что решение существует. Часто эту проблему удается решить, изменив некоторые параметры и повторно запустив *Поиск решения*. Указанные параметры устанавливаются в ДО *Параметры*, которое отобразится, если в ДО *Параметры поиска решения* выбрать кнопку *Параметры*.

Пример – Решить систему уравнений

$$\begin{cases} x + y + z = 6; \\ 2x - y + z = 3; \\ x - 4y + 3z = 2. \end{cases}$$

Решим задачу встроенными вычислительными инструментами Excel для анализа данных *Поиск решения*.

1 Добавьте новый лист и присвойте ему имя Система.

2 В ячейки В1, В2 и В3 внесите начальные приближения переменных, например, 0. Это будут изменяемые ячейки.

3 В столбец D (целевая ячейка) внесите правые части уравнений, ссылаясь на введенные значения, т. е. ячейки В1, В2 и В3.

4 В столбец Е внесите правые (известные) части уравнений. В результате должно получиться, как на рисунке 4.

5 Выберите команду меню Данные \rightarrow Анализ \rightarrow Поиск Решения.

6 Заполните окно *Поиск решения*, как показано на рисунке 5. Одно из введенных уравнений, например D1, принимается за целевую функцию.

	А	В	С	D	Е	
1	х	0		=B1+B2+B3	6	
2	у	0		=2*B1-B2+B3	3	
3	z	0		=B1-4*B2+3*B3	2	
	1					

Рисунок 4 – Заполнение ячеек данными для поиска решения

Установить целевую ячейку:	<u>В</u> ыполнить
Равной: О максимальному значению	Закрыть
Измендя ячейки:	
Ограничения:	<u>П</u> араметры
\$D\$2:\$D\$3 = \$E\$2:\$E\$3 Л	
Изменить	Восс <u>т</u> ановить
– <u>Уд</u> алить	Справка

Рисунок 5 – Заполнение ДО Поиск решения

7 Нажав кнопку *Выполнить*, в ячейках B1, B2 и B3 Вы увидите результат вычислений (рисунок 6).

	А	В	С	D	Е	F
1	х	1		6,000001	6	
2	у	2		3	3	
3	Z	3,000001		2	2	

Рисунок 6 – Результаты поиска решений

Задание

Решить уравнение и систему уравнений согласно варианту, полученному у преподавателя.

Контрольные вопросы

1 Для чего используются надстройки?

2 Какие надстройки есть в Excel?

3 Что позволяет функция Подбор параметра?

4 В каких случаях применяется функция Поиск решений?

9 Лабораторная работа № 9. Алгоритмический язык VBA. Ознакомление с интегрированной системой программирования

Цель работы: ознакомление с основными окнами среды VBA и с процессом конструирования визуального проявления программы.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить отчет в виде набора файлов.

Основные теоретические положения

Корпорация Microsoft интегрировала в свои офисные продукты язык программирования Visual Basic for Applications (VBA). С помощью этого языка каждый пользователь может автоматизировать работу приложения и максимально приспособить его работу для решения текущих задач.

Код VBA набирается в редакторе Visual Basic. Для того чтобы попасть в этот редактор, даётся команда *Сервис – Макрос – Редактор Visual Basic* или вводится комбинация клавиш *Alt* + *F11*. В результате откроется интегрированная среда разработки приложений (IDE).

Возвратиться из редактора Visual Basic в рабочую книгу можно, нажав кнопку с пиктограммой Excel.

Все приложения, написанные на VBA, создаются как проекты. В проект входят несколько модулей: код программы, параметры формы, конфигурация интегрированной среды разработки и др.

Среда разработки приложений имеет стандартный для Windows-приложений вид: строка меню, панель инструментов и еще несколько открытых окон. На рисунке 7 открыты два окна: *Project – VBA Project и Properties*.

Окно Project – VBA Project активизируется выбором команды View – Project Explorer (1) или нажатием кнопки Project Explorer (2) панели инструментов. В окне Project – VBAProject представлена иерархическая структура форм и модулей текущего проекта. В этом окне отображается реестр модулей и форм, входящих в создаваемый проект. Двойным щелчком на значке модуля в окне Project – VBA Project можно открыть соответствующий модуль.

При написании кода программы необходимо придерживаться следующей структуры:

Sub ИмяПрограммногоМодуля () Объявление переменных и констант

Тело программы (последовательность исполняемых операторов) EndSub

Переменные в программе используются для хранения данных, которые могут изменяться в процессе выполнения процедуры. Объявление типа переменной означает, что пользователь устанавливает определённые границы, в которых может изменяться переменная. Тип переменной можно вообще не определять. Если тип переменной не объявляется, по умолчанию он принимается как тип Variant.

街 Microsoft Visual Basic	- Построение_UserForm.xl	s - [Module1 (Code)]	×
Eile Edit View	Insert Format Debug	<u>R</u> un <u>T</u> ools <u>A</u> dd-Ins <u>W</u> indow <u>H</u> elp _	θ×
i 🛛 🖳 - 🕞 🕹 🗗	◎ 舟 ! ウ で ▶ Ⅱ		
Project - VBAProject	×	(General) (Declarations)	
	Ŧ		
🗄 😻 SOLVER (SOLVER.)	(LAM)		-
🖻 💩 VBAProject (Пост	роение_UserForm.xls)		
Microsoft Excel C	Objects		
Пист1 (Лист	1)		
Ш ЛИСТ 2 (ЛИСТ Ш Лист 2 (ЛИСТ	2)		
] ЭтаКымга	3)		
Module1			
Descention durant	v 1		
Properties - Jucit			
Лист1 Worksheet			
Alphabetic Categorized			
(Name)	Лист1		
DisplayPageBreaks	False		
DisplayRightToLeft	False		
EnableAutoFilter	False		
EnableCalculation	True		
EnableFormatConditionsCale	True		
EnableOutlining	False		
EnablePivotiable	raise		
Name	U - XINORESUICTIONS		
ScrollArea	70%-11		
StandardWidth	8.43		
Visible	-1 - xlSheetVisible		_
			N

Рисунок 7 – Интегрированная среда разработки приложений (IDE)

В таблице 1 содержится информация о размере данных, т. е. об объёме памяти, выделяемом для хранения данных.

Тип	Содержание	Объем	Диапазон значений
Byte	Короткое неотрицательное	1Б	0255
	число		
Integer	Целое число	2Б	-32768+32767
Long	Длинное целое число	4Б	-2147483648 +2147483647
Single	Десятичное число обычной	4Б	$\pm (1,4011298e - 453,402823e + 38)$
	точности		
Double	Десятичное число двойной	8Б	$\pm(1e-3241e+308)$
	точности		
String	Набор символов (строка)	Зависит от числа символов в строке	
Boolean	Логическая величина	2Б	Истина (True) или Ложь (False)
Date	Дата	8Б	Информация о дате
Object	Указатель на объект	4Б	Значением является ссылка на объект
Variant	Произвольное значение	Не ме-	Может быть переменной любого типа
		нее 16 Б	

Таблица 1 – Основные типы переменных языка VBA

Для объявления переменной используется оператор Dim. Этот оператор имеет следующий синтаксис:

Dim ИмяПеременной As ТипДанных

Имена переменных должны начинаться с буквы, могут содержать также цифры и знаки подчёркивания. Имя не может содержать пробелы, точки, запятые, восклицательные знаки и символы @, &, \$, # и не должно иметь более 255 символов.

Если в вычислениях нужна величина, которая бы не меняла своего значения, то применяются константы. Для их объявления используется оператор Const, имеющий следующий синтаксис:

Const ИмяКонстанты As TunДaнных = Значение

Например:

Const GruppaAsInteger = 25

Для задания переменным исходных значений используется функция InputBox, а для вывода значений переменных на экран – процедура MsgBox. Синтаксис их написания можно посмотреть в справочной системе (F1).

Для вычисления значений переменных в программе используется оператор присваивания, имеющий следующий синтаксис:

Переменная = Выражение

Если *Выражение* математическое, то в нем используются знаки арифметических действий, математические функции и круглые скобки, например:

$$A = 2 \times \sin(x+2) / \cos(x-2)$$

Задание

1 Составить необходимый код программы с обеспечением ввода исходных данных и вывода результата на экран.

2 Запустить программу на выполнение и сравнить полученный результат со значением, вычисленным, например, с помощью калькулятора.

Контрольные вопросы

1 Как объявить переменную, которая может иметь целые значения в диапазоне от 0 до 100?

2 Как в программе присвоить значение переменной?

3 Как показать результат вычислений?

10 Лабораторная работа № 10. Алгоритмический язык VBA. Разработка разветвленной программы с использованием оператора IF

Цель работы: изучение оператора ветвления *If*; создание программы с использованием оператора ветвления *If* на языке программирования VBA.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить отчет в виде набора файлов.

Основные теоретические положения

Алгоритм называется разветвляющимся, если последовательность выполнения его шагов изменяется в зависимости от выполнения некоторых условий. Условие – это логическое выражение, которое может принимать одно из двух значений: «ДA» – если условие верно (истинно, *TRUE*); «*HET*» – если условие неверно (ложно, *FALSE*).

Схема алгоритма условного оператора *If* представлена на рисунке 8.



Рисунок 8 – Схема алгоритма конструкции условного оператора *If*

Синтаксис условного оператора *If* в однострочной форме следующий:

If<лог. выраж.>*ThenP*₁:*P*₂:...:*P*_N*ElseM*₁: *M*₂: ... *M*_N

Возможна и другая синтаксическая форма – блочная (структурная):

If<логическое выражение> Then P1 ... PN Else M1 M_N End If

где *If*, *Then*, *Else*, *End If* – зарезервированные слова; $P_1, P_2, P_N, M_1, M_2, M_N$ – операторы.

Задание

- 1 Запросить у пользователя ввод числа.
- 2 Сравнить введенное число с другим заданным числом, например 20.
- 3 По результатам сравнения вывести соответствующее сообщение:

«25 > 20» или «15 < 20»,

где 25, 15 – введенные пользователем числа.

Контрольные вопросы

1 Можно ли вставлять инструкцию *Else* перед инструкцией *ElseIf* в блочном варианте оператора ветвления?

2 В каких случаях в программе используется полный условный оператор?

- 3 Как оформляется полный условный оператор?
- 4 Как он работает (что происходит при его выполнении)?
- 5 Когда в программе используется неполный условный оператор?

11 Лабораторная работа № 11. Алгоритмический язык VBA. Разработка разветвленной программы с использованием оператора выбора

Цель работы: ознакомление с оператором выбора *SelectCase* и закрепление полученных знаний на практике.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить отчет в виде набора файлов.

Основные теоретические положения

При наличии большого количества ветвлений конструкция многозначных ветвлений *If* становится тяжёлой для восприятия. В подобных случаях хорошей альтернативой оператору *If* служит оператор выбора *SelectCase*, который позволяет выбрать одно из нескольких возможных продолжений программы.

В то время как *If* ... *Then* ... *Else* для каждой инструкции *ElseIf* оценивает разные выражения, инструкция *SelectCase* оценивает выражение только один

раз, в начале управляющей структуры. *SelectCase* выполняет одну из нескольких групп инструкций в зависимости от значения выражения. Синтаксис:

SelectCase<выражение> [Case <cписокВыражений-n> [инструкции-n]] ... [CaseElse [инструкции_else]] EndSelect

где *<выражение>* – обязательный, любое числовое выражение или строковое выражение; *<списокВыражений*-n> – обязательный при наличии предложения Case. Список с разделителями, состоящий из одной или нескольких форм следующего вида:

a) *<инструкции-n>* – необязательный. Одна или несколько инструкций, выполняемых в том случае, если выражение совпадает с любым компонентом списка *<списокВыражений-n>*;

б) *<инструкции_else>* – необязательный. Одна или несколько инструкций, выполняемых в том случае, если выражение не совпадает ни с одним из предложений *Case*.

Синтаксис оператора *SelectCase* также может содержать следующие элементы:

- выражение То выражение;

- *Is* оператор сравнения *выражение*.

Ключевое слово *To* задает диапазон значений. При использовании ключевого слова *To* перед ним должно находиться меньшее значение. Ключевое слово *Is* с операторами сравнения (кроме *Is* и *Like*) задает диапазон значений. Если ключевое слово *Is* не указано, оно вставляется по умолчанию.

Если выражение совпадает с любым выражением из списка Выражений в предложении Case, выполняются все инструкции, следующие за данным предложением Case до следующего предложения Case, или для последнего предложения до инструкции End Select. Затем управление передается инструкции, следующей за End Select. Если выражение совпадает с выражениями из списка в нескольких предложениях Case, выполняется только первый подходящий набор инструкций.

Предложение *CaseElse* задает список *инструкции_else*, которые будут выполнены, если не обнаружено ни одно совпадение выражения и компонента *список Выражений* ни в одном из остальных предложений *Case*.

Хотя данное предложение не является обязательным, рекомендуется помещать предложение *CaseElse* в блок *SelectCase*, чтобы предусмотреть неожиданные значения выражения. Если ни в одном предложении *Case список Выражений* не содержит компонента, отвечающего аргументу выражения, и отсутствует инструкция *CaseElse*, выполнение продолжается с инструкции, следующей за инструкцией *EndSelect*.

Пример использования оператора *SelectCase* представлен в таблице 2.

Синтаксис оператора Select Case	Пример использования оператора SelectCase
SelectCase Ключ Выбора	SelectCase vozrast
Case Is выражение	Case Is $\leq =7$
onepamop	Msgbox «Ты дошкольник»
Case диапазон значений	<i>Case 8 to 16</i>
onepamop	Msgbox «Ты учишься в школе»
Case диапазон значений	<i>Case 17 to 30</i>
onepamop	Msgbox «Тебе пора заняться делом»
Case диапазон значений	Case 31 to 60
onepamop	Msgbox «Кто не работает, тот не ест»
CaseElse	CaseElse
onepamop	Msgbox «Вы заслужили отдых»
End Select	EndSelect

Таблица 2 – Синтаксис и пример использования оператора SelectCase

Если значение переменной *vozrast* меньше или равно 7, отображается сообщение «*Ты дошкольник*»; если значение находится в диапазоне от 8 до 16 – сообщение «*Ты учишься в школе*»; если в диапазоне от 17 до 30 – сообщение «*Тебе пора заняться делом*»; если в диапазоне от 31 до 60 – сообщение «*Кто не работает, тот не ест*». Если значение возраста не равно ни одному из предложенных диапазонов значений, выводится сообщение «*Вы заслужили отдых*».

Из представленного в таблице 2 примера видно, что код этой процедуры более прост для восприятия, чем многозначные ветвления *If*.

Контрольные вопросы

1 В каких случаях в программе используется оператор выбора?

2 В чем преимущество оператора выбора варианта перед многовариантным оператором ветвления?

3 Когда применение оператора *SelectCase* эффективнее оператора *IfThenElseEndIf*?

12 Лабораторная работа № 12. Алгоритмический язык VBA. Разработка программы с использованием оператора выбора FOR. Ввод и вывод одномерного массива

Цель работы: изучение оператора цикла For.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить отчет в виде набора файлов.

Основные теоретические положения

Часто в программах необходимо реализовать определённые операторы несколько раз. В этих случаях организуют циклические вычисления. Алгоритм называется *циклическим*, если определенная последовательность шагов выполняется несколько раз в зависимости от заданных условий. Циклические алгоритмы могут быть осуществлены с применением следующих операторов цикла: *For ... Next, While ... Wend, Do ... Loop*, которые позволяют повторить группу операторов или один оператор заданное количество раз.

Общий вид алгоритма конструкции оператора цикла *For ... Next* представлен на рисунке 9.



Рисунок 9 – Схема алгоритма конструкции оператора цикла For ... Next

Синтаксис конструкции оператора цикла For ... Next следующий:

For
$$I = N_1$$
 To N_2 [Step h]
 P_1
...
[Exit For]
 P_N
Next i

где *For* (для), *To* (до), *Step* (шаг), *Exit For* (выход из *For*), *Next* (следующий) – служебные слова;

VBA, *P1*, *PN* – операторы.

Step является необязательным параметром. Если он опущен в программе, то значение параметра *i* увеличивается на 1. Параметр *Step* может быть любым действительным числом: как целым, так и дробным, как положительным, так и отрицательным.

Оператор *Exit For* позволяет выйти из цикла *For...Next* до его завершения. Тем самым программа сможет среагировать на определённое событие, не выполняя цикл заданное число раз.

Контрольные вопросы

1 Может ли тело оператора цикла с параметром не выполниться ни разу?

2 Как должен быть оформлен оператор цикла с параметром, чтобы тело цикла выполнялось при уменьшающихся значениях параметра цикла?

3 Чему равно количество повторений тела оператора цикла с параметром, если параметр цикла принимает:

а) все целые значения от 1 до 10;

б) все значения от 10 до 100 с шагом 7;

в) все значения от а до б с шагом *Step*.

4 Можно ли в теле оператора цикла использовать условный оператор?

5 Какие Вы знаете операторы для принудительного (преждевременного) выхода из оператора цикла?

13 Лабораторная работа № 13. Алгоритмический язык VBA. Обработка одномерного массива

Цель работы: изучение одномерных массивов; разработка программы с вводом, выводом и обработкой одномерных массивов.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить отчет в виде набора файлов.

Основные теоретические положения

Массив – совокупность однотипных элементов данных (чисел, логических данных, символов), которой при обработке присвоено определенное имя. Совокупность представлена набором переменных с одним именем и с разными индексами. Массивы бывают разной размерности: одномерные, двумерные, трехмерные, ..., *n*-мерные.

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<mun элементов массива>]

или

Dim<имя массива>(<количество элементов массива>) [As<mun 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**<mun 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 **OptionBase 1**.

Пример.

DimA(1 To 10) As Integer – массив А состоит из 10 элементов целого типа, индексы которых 1, 2, ..., 10.

Dim A(10) As Integer – массив состоит из 10 значений целого типа. Индексация зависит от опции **Base**. Если опция не указана, то номера элементов – от 0 до 9, если же указана (т. е. вначале модуля записано **OptionBase 1**), то номера элементов изменяются от 1 до 10;

3) динамический массив:

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

После определения количества элементов массива выполняется его переопределение:

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

Пример.

DimA() As Single – динамический массив A вещественных элементов n = 7;

ReDimA (1 To n) – переопределение одномерного массива из *n* значений.

ReDimA (5, n) – переопределение двумерного динамического массива, состоящего из пяти строк и n столбцов (начало индексации элементов определяется по опции **Base**).

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

Примеры объявления массивов:

DimA (12) As Integer одномерный массив из 12 элементов типа Integer *Dim A(1 To 12) As Integer*

Dim B(1 To 3, 1 To 3) As *Single*

Причем по умолчанию первый элемент массива A будет A(0), а последний – A(11). В этом случае говорят, что 0 – базовый индекс. Можно изменить базовый индекс, написав в начале листа модуля инструкцию **OptionBase 1**. После этого индексы массивов A и B будут начинаться с единицы.

Массив в программе определяется поэлементно.

Удобным способом определения одномерных массивов является функция Array, преобразующая список элементов, разделенных запятыми, в вектор из этих значений и присваивающая их переменной тип Variant.

Задание

1 Объявить одномерный массив размерностью 10 элементов целочисленного типа.

2 Инициализировать все элементы массива произвольными числами: A(1) = 5 и т. д.

3 Реализовать расчет среднего арифметического всех чисел, содержащихся в массиве, с помощью оператора цикла.

4 Вывести результат расчета в виде текстового сообщения:

«Среднее арифметическое: _____».

Контрольные вопросы

1 Опишите синтаксис объявления массива с использованием оператора **Dim**.

2 Какое значение нижнего индекса элемента массива принято в VBA по умолчанию? Каким образом можно для него задать значение 1?

3 Как установить или изменить размерность многомерного динамического массива?

4 Как изменить размерность динамического массива без потери имеющихся значений его элементов?

5 Сколько индексов характеризуют конкретный элемент двумерного массива?

14 Лабораторная работа № 14. Алгоритмический язык VBA. Разработка программы с вложенными циклами. Обработка многомерных массивов

Цель работы: изучение вложенных операторов цикла For.

Порядок выполнения работы

1 Ознакомиться с основными теоретическими положениями.

2 Выполнить задание, полученное у преподавателя.

3 Оформить отчет в виде набора файлов.

Основные теоретические положения

Если телом цикла является циклическая структура, то такие циклы называются вложенными. Цикл, содержащий в себе другой цикл, – внешний, а цикл, содержащийся в теле другого цикла, – внутренний.

Синтаксис конструкции вложенных операторов цикла For ... Next:

For
$$i = N_1$$
 To N_2
For $j = M_1$ To M_2
 P_1
...
 P_N
Next j
Next i
 $meno$ внутреннего
цикла
 $meno$ внешнего
цикла

Общий вид алгоритма конструкции вложенных операторов цикла *For* ... *Next* представлен на рисунке 10.



Рисунок 10 – Схема алгоритма конструкции вложенных операторов цикла For ... Next

При первом вхождении в цикл параметр внешнего цикла i принимает значение, равное N_1 , и управление передаётся во внутренний цикл, в котором параметр цикла j принимает значение, равное M_1 , и выполняется оператор (операторы), который записан во внутреннем цикле. Затем параметр внутреннего цикла j увеличивается на 1 и вновь повторяется тело цикла.

Операторы P_1 , P_N будут реализовываться до тех пор, пока параметр цикла i не станет больше величины M_2 . Затем параметр внешнего цикла i увеличивается на 1 и вновь начинает свою работу внутренний цикл, в котором параметр цикла j будет изменяться от M_1 до M_2 , и при каждом прохождении цикла будут выполняться операторы P_1 и P_N . Внешний цикл закончит свою работу, когда параметр цикла i станет больше величины N_2 .

Задание

С помощью вложенных операторов цикла *For* ... *Next* реализовать запись на рабочий лист *Excel* матрицы размерностью $n \times n$ (значение числа n запросить у пользователя), элементы которой должны являться произведением номера строки на номер колонки.

Контрольные вопросы

1 В каких случаях используются вложенные операторы цикла?

2 Как оформляются вложенные операторы цикла?

3 Нарисуйте алгоритмическую схему выполнения вложенных операторов цикла.

4 Вложенный цикл образован двумя операторами цикла с параметром. Что является телом?

5 Может ли внешний оператор вложенного цикла:

а) не выполниться ни разу;

б) выполняться бесконечное число раз (или до того момента, когда пользователь прервет его выполнение)?

15 Лабораторная работа № 15. Алгоритмический язык VBA. Разработка программы с использованием циклов While

Цель работы: получение навыков применения операторов цикла с предусловием и постусловием.

Порядок выполнения работы

1 Ознакомиться с основными теоретическими положениями.

2 Выполнить задание, полученное у преподавателя.

3 Оформить отчет в виде набора файлов.

Основные теоретические положения

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

Циклы с предусловием (DoWhile ... Loop, DoUntil ... Loop) представлены в таблице 3, а операторы циклов с постусловием (Do ... LoopWhile, Do ... LoopUntil) – в таблице 4.

Синтаксис	DoWhile < <i>yсловие</i> >	DoUnlil <условие>
	<тело цикла>	<тело цикла>
	[ExitDo]	[ExitDo]
	Loop	Loop
Порядок	<Тело цикла> будет выполняться в том	<Тело цикла> выполняется до
выполнения	случае, когда <условие> имеет значение	тех пор, пока <i><условие></i> не
	Истина (TRUE) (цикл продолжается при	примет значение Истина (цикл
	истинном значении <i><условия></i>). Если	продолжается при ложном зна-
	<i><условие></i> ложно (FALSE), то выполня-	чении <условия>). Есть воз-
	ются операторы, стоящие за циклом. В	можность досрочного выхода
	первом случае есть возможность досроч-	из цикла (это реализовано через
	ного выхода из цикла (это реализовано	ExitDo)
	через ExitDo)	
Изображение		
в блок-схемах		
	SVCIOBUE> HET	Да
		нет
	<тело цикла>	<тело цикла>

Таблица 3 – Циклы с предусловием

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.

Пример – Организовать ввод последовательности целых чисел, пока их сумма не превысит целого числа т. Вывести количество введенных чисел. Текст программы с пояснениями представлен в таблице 5.

Задание

- 1 В цикле реализовать запрос пароля: «Введите пароль:».
- 2 Сравнить введенный пользователем пароль с заданным.
- 3 Цикл должен выполняться до тех пор, пока пароль не совпадет.
- 4 Если пароль совпадает, цикл завершается и программа что выдает?

Синтаксис	Do	Do
	<телоцикла>	<тело цикла>
	[Exit Do]	[Exit Do]
	Loop While<условие>	Loop Until < yсловиe>
Порядок	<Тело цикла> будет выполняться в том слу-	<Тело цикла> выполняется до
выполнения	чае, когда <условие> имеет значение Исти-	тех пор, пока <i><условие></i> не при-
	на (цикл продолжается при истинном зна-	мет значение истина (цикл про-
	чении <i><условия></i>). Если <i><условие></i> ложно,	должается при ложном значе-
	то выполняются операторы, стоящие за	нии <i><условия></i>). Есть возмож-
	циклом. Предоставлена возможность до-	ность досрочного выхода из
	срочного выхода из цикла (это реализовано	цикла (это реализовано через
	через ExitDo)	ExitDo)
Изображе-		
ние в блок-		
схемах	<тело цикла>	<тело цикла>
	да <условие>	нет
	June Her	<условие>
	nei	Тда

Таблица 4 – Циклы с постусловием

Таблица 5 – Программа по вводу последовательности целых чисел

Текст программы	Описание
Public Sub prog1()	_
Dim x As Integer, m As Integer	-
Dim s As Integer, i As Integer	—
т = InputBox(«Введите число т»)	Ввод предельного числа
i = 1	Номер вводимого числа последовательности
s =InputBox(«Введите 1 число»)	Ввод первого числа последовательности
Do While $s \le m$	Цикл с предусловием: тело цикла выполняется, пока
	условие s<=т имеет значение Истина (TRUE)
i = i + 1	Тело цикла: увеличение номера на 1
x=InputBox(«Введите» &i& «число»)	ввод очередного (<i>i</i> -го) значения
s = s + x	добавление введенного значения к предыдущему
	значению суммы
Loop	Конец тела цикла
MsgBox («Количество чисел» &i)	Вывод значения переменной і
EndSub	

Контрольные вопросы

1 В каких случаях используются операторы цикла с условием?

2 В чём основное отличие циклов с предусловием и с постусловием?

3 В каких случаях целесообразно использовать циклы с предусловием, циклы с постусловием и циклы по счётчику?

4 Может ли тело оператора цикла с предусловием:

а) не выполниться ни разу;

б) выполняться бесконечное число раз (или до тех пор, когда пользователь прервет его выполнение)?

16 Лабораторная работа № 16. Работа в MathCad. Использование пакета MathCad для решения математических задач

Цель работы: ознакомление с методами решения алгебраических уравнений в MathCad.

Порядок выполнения работы

1 Ознакомиться с основными теоретическими положениями.

2 Выполнить задание, полученное у преподавателя.

3 Оформить отчет в виде набора файлов.

Основные теоретические положения

Для алгебраических уравнений вида f(x) = 0 решение в MathCad находится с помощью функции *root*.

Общий вид функции следующий:

root(f(x), x),

где $f(x) - \phi y h \kappa u g x$, описывающая левую часть выражения вида f(x) = 0;

х – имя *переменной*, относительно которой решается уравнение.

Функция *root* реализует алгоритм поиска корня численным методом и требует предварительного задания начального приближения искомой переменной *x*. Поиск корня будет производиться вблизи этого числа. Таким образом, присвоение начального значения требует предварительной информации о примерной локализации корня.

Если после многих итераций MathCad не находит подходящего приближения, то появится сообщение «отсутствует сходимость».

Эта ошибка может быть вызвана следующими причинами:

1) уравнение не имеет корней;

2) корни уравнения расположены далеко от начального приближения;

3) выражение f(x) имеет разрывы между начальным приближением и корнем;

4) выражение имеет комплексный корень, но начальное приближение было вещественным и наоборот.

В зависимости от типа задачи функция *root* может включать либо два, либо четыре аргумента и, соответственно, использует разные алгоритмы поиска корней.

root(f(x),x); root (f (x) , x, a, b), где f (x) – скалярная функция, определяющая уравнение f(x) = 0;

Х – имя скалярной переменной, относительно которой решается уравнение;

a, b – границы интервала, внутри которого происходит поиск корня.

Первый тип функции *root* требует дополнительного задания начального значения переменной *x* (примерная локализация корня, поиск корня будет производиться вблизи этого числа).

Для задания начального значения или интервала удобно предварительно построить график функции. Чем точнее выбрано начальное приближение корня, тем быстрее будет *root* сходиться.

Пример – Найти корень уравнения $sin(2x) + cos(2x) = \sqrt{x} sin(3x)$ при заданном начальном значении x = 5. Выполнить графическую интерпретацию результата.

Реализация в MathCad:

1) привести уравнение к виду f(x) = 0, если это необходимо, например, уравнение $\sin(2x) + \cos(2x) = \sqrt{x} \sin(3x)$ преобразуется в $\sin(2x) + \cos(2x) - \sqrt{x} \sin(3x) = 0$;

2) установить курсор в свободное место рабочего окна документа MathCad;

3) определить функцию, содержащуюся в левой части уравнения вида f(x) = 0, например $f(x) := \sin(2x) + \cos(2x) - \sqrt{x} \sin(3x)$;

4) построить график функции f(x) (рисунок 11);

5) установить оси графика в виде креста;

6) задать начальное приближение искомой переменной *x*;

7) подставляя в качестве параметров функции *root* имя функции левой части уравнения и переменную, содержащую начальное приближение, вывести значение корня уравнения при заданном начальном приближении с помощью оператора «=»;

8) задать аргумент функции f(x) в виде дискретной переменной, при этом диапазон изменения аргумента должен включать найденный корень уравнения.



Рисунок 11 — Пример построения графика f(x)

Задание

1 Ознакомиться с методом решения алгебраических уравнений с помощью функции *root* и выполнить приведённый выше пример.

2 С помощью функции *root* решить уравнения:

a)
$$\cos(x) - x - 0, 2 = 0$$
;

6)
$$1, 2 \cdot x^3 - 10 \cdot x + 2 = 0$$
;

B)
$$4 \cdot x^4 - 10 \cdot x^3 + 2 \cdot x - 1 = 0$$
.

Контрольные вопросы

1 Какие функции имеются в пакете MathCad для решения уравнений?

2 Какие алгоритмы поиска корней уравнений реализует функция root?

3 С какой целью желательно предварительно построить график функции?

4 По каким причинам может появиться сообщение «отсутствует сходимость»?

Список литературы

1 **Гвоздева, В. А.** Информатика, автоматизированные технологии и системы: учебник / В. А. Гвоздева. – Москва: ФОРУМ; ИНФА-М, 2022. – 542 с.

2 Гуриков, С. Р. Информатика: учебник / С. Р. Гуриков. – 2-е изд., перераб. и доп. – Москва: ФОРУМ; ИНФА-М, 2022. – 630 с.

3 Информатика. Базовый курс: учебное пособие / Под ред. С. В. Симоновича. – 3-е изд. – Санкт-Петербург: Питер, 2017. – 640 с.: ил.

4 **Гуриков, С. Р.** Интернет-технологии: учебное пособие / С. Р. Гуриков. – Москва: ФОРУМ; ИНФРА-М, 2017. – 184 с.

5 Гуриков, С. Р. Введение в программирование на языке Visual Basic for Applications (VBA): учебное пособие / С. Р. Гуриков. – Москва: ИНФРА-М, 2020. – 317 с.

6 Скитер, Н. Н. Информационные технологии: учебное пособие / Н. Н. Скитер, А. В. Костикова. – Волгоград: ВолгГТУ, 2019. – 96 с.