

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

ИНФОРМАТИКА

*Методические рекомендации к лабораторным работам
для студентов специальности*

*1-53 01 01 «Автоматизация технологических процессов и
производств (по направлениям)» очной и заочной форм обучения*

Часть 2



Могилев 2023

УДК 004
ББК 32.81
И74

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления»
«08» декабря 2022 г., протокол № 5

Составитель ст. преподаватель Т. Л. Шебан

Рецензент канд. техн. наук, доц. С. К. Крутолевич

Методические рекомендации к лабораторным работам для студентов специальности 1-53 01 01 «Автоматизация технологических процессов и производств (по направлениям)» очной и заочной форм обучения.

Учебное издание

ИНФОРМАТИКА

Часть 2

Ответственный за выпуск	А. И. Якимов
Корректор	И. В. Голубцова
Компьютерная верстка	М. М. Дударева

Подписано в печать 18.08.2023 . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. 1,16. Уч.-изд. л. 1,25. Тираж 21 экз. Заказ № 881.

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, 2023

Содержание

1 Лабораторная работа № 9. Алгоритмический язык VBA	4
2 Лабораторная работа № 10. Алгоритмический язык VBA	5
3 Лабораторная работа № 11. Алгоритмический язык VBA	6
4 Лабораторная работа № 12. Алгоритмический язык VBA	7
5 Лабораторная работа № 13. Алгоритмический язык VBA	8
6 Лабораторная работа № 14. Алгоритмический язык VBA	10
7 Лабораторная работа № 15. Алгоритмический язык VBA	12
8 Лабораторная работа № 16. Алгоритмический язык VBA	14
9 Лабораторная работа № 17. Алгоритмический язык VBA	15
10 Лабораторная работа № 18. Использование пакета Mathcad для решения математических задач	17
11 Лабораторная работа № 19. Использование пакета Mathcad для решения математических задач	19
Список литературы	20

1 Лабораторная работа № 9. Алгоритмический язык VBA

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.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить и представить отчет в виде файла в формате документа Word. Отчет должен содержать название работы, цель работы, задание, схему алгоритма, код программы.

Основные теоретические положения

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 A1 (колонка – буква – строка – число).

Например, выражение $X = Worksheets(«Лист1»).Range(«B1»).Value$ присваивает переменной *X* значение ячейки «B1» листа «Лист1».

Выражение $Worksheets(«Лист1»).Range(«B7:C9»).Value = 3$ выводит в диапазон ячеек «B7:C9» 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 *A* присваивает значение из ячейки первой строки и второго столбца первого листа,

Выражением $Worksheets(1).Cells(2,2).Value = X$ в ячейку второй строки и второго столбца заносится значение переменной *X*.

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

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

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

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.

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

где `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;

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

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

Контрольные вопросы

- 1 Назовите способы ввода и вывода данных.
- 2 Назовите основные типы данных VBA.

2 Лабораторная работа № 10. Алгоритмический язык VBA

Цель работы: изучение оператора условия IF.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить и представить отчет в виде файла в формате документа Word. Отчет должен содержать название работы, цель работы, задание, схему алгоритма, код программы.

Основные теоретические положения

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 [операторы 1] [Else операторы 2]

- 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, то «операторы 2».

В представленном ниже примере определяется, является ли введенное число четным.

```
Public Sub chet()
Dim n As Double
n = InputBox("Введите число")
If n Mod 2 = 0 Then
MsgBox ("Введенное число " & n & " является четным")
Else
MsgBox ("Введенное число " & n & " не является четным")
End If
End Sub
```

Контрольные вопросы

- 1 Какие виды оператора IF существуют? Чем они отличаются друг от друга?
- 2 Сколько условий может проверять оператор IF блочной формы записи?

3 Лабораторная работа № 11. Алгоритмический язык VBA

Цель работы: изучение оператора цикла с параметром For...Next.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить и представить отчет в виде файла в формате документа Word. Отчет должен содержать название работы, цель работы, задание, схему алгоритма, код программы.

Основные теоретические положения

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

```
For <счетчик цикла>=<начало> To <конец> [Step шаг]
<операторы>
[Exit For]
Next <счетчик цикла>,
```

где <счетчик цикла> – любая числовая переменная, в которой сохраняется информация о количестве выполненных проходов цикла;

параметры <начало> и <конец> – это числовые выражения, задающие начальное и конечное значение счетчика и определяющие количество проходов цикла.

Числовая переменная шаг задает приращение, на которое увеличивается счетчик цикла при каждом проходе. [Step шаг] необязательная фраза, при её отсутствии VBA увеличивает счетчик цикла на единицу.

Работает оператор следующим образом. Переменной <счетчик цикла> присваивается значение параметра <начало> и проверяется условие счетчик цикла меньше <конец>. Если условие неверно, то управление передается оператору, следующему за ключевым словом Next. Если же условие верно, то выполняются все операторы, входящие в блок, обозначенный как <операторы>, вплоть до ключевого слова Next. После этого значение переменной <счетчик цикла> увеличивается на значение, заданное во фразе [Step шаг] или на единицу, если фраза опущена, и управление возвращается на ключевое слово For. Данный процесс будет выполняться, пока значение <счетчик цикла> не достигнет значения <конец>. Досрочно завершить цикл For Next можно с помощью оператора Exit For.

Например, следующий код программы запрашивает количество чисел для ввода, выводит заданное количество чисел из диапазона от –5 до 5 в первый столбец листа Excel, а числа, большие нуля, выводит во второй столбец.

```
Public Sub param()
Dim n As Double
n = InputBox("Введите количество чисел")
For i=1 to n
Cells(i,1)=Int(rnd()*10-5)
If Cells(i,1)>0 Then
Cells(i,2)= Cells(i,1)
End If
Next
End Sub
```

Контрольные вопросы

- 1 В каких случаях используется оператор цикла с параметром?
- 2 Может ли шаг изменения переменной цикла быть отрицательным?

4 Лабораторная работа № 12. Алгоритмический язык VBA

Цель работы: освоение основных способов обработки массивов.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить и представить отчет в виде файла в формате документа Word. Отчет должен содержать название работы, цель работы, задание, схему алгоритма, код программы.

Основные теоретические положения

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 – это массивы, в которых либо неизвестно начальное количество элементов, либо размерность массива (количество элементов) изменяется при выполнении программы.

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.

В представленном ниже примере объявляется двумерный динамический массив, который после ввода размерности переопределяется в соответствии с введенным количеством строк и столбцов. Далее массив заполняется случайными числами от -10 до 10 и выводится на лист Excel.

```
Public Sub massiv()
Dim A() As Double
n = InputBox("Введите количество строк массива")
m = InputBox("Введите количество столбцов массива")
ReDim A(1 To n, 1 To m)
For i = 1 To n
  For j = 1 To m
    A(i, j) = Int(Rnd() * 20 - 10)
    Cells(i, j) = A(i, j)
  Next
Next
End Sub
```

Контрольные вопросы

- 1 Какие виды массивов существуют? Чем они отличаются друг от друга?
- 2 Назовите способы заполнения массива.

5 Лабораторная работа № 13. Алгоритмический язык VBA

Цель работы: освоение основных приемов применения операторов цикла с предусловием и постусловием.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.

3 Оформить и представить отчет в виде файла в формате документа Word. Отчет должен содержать название работы, цель работы, задание, схему алгоритма, код программы.

Основные теоретические положения

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

Операторы циклов с предусловием: Do While...Loop, While...Wend, Do Until...Loop. Синтаксис операторов циклов с предусловием представлен в таблице 1.

Операторы циклов с постусловием: Do...Loop While, Do...Loop Until. Синтаксис операторов циклов с постусловием представлен в таблице 2.

Таблица 1 – Синтаксис операторов циклов с предусловием

Синтаксис	Do While <условие> <тело цикла> [Exit Do] ... Loop	While <условие> <тело цикла> Wend	Do Until <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)

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 2 – Операторы циклов с постусловием

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)

Представленный ниже пример демонстрирует использование цикла с предусловием. В этом примере числа вводятся, пока не встретится три подряд идущих положительных числа, после чего выводится сообщение об общем количестве введенных чисел.

```

Public Sub chet()
Dim n As Double
Dim kolp As Double
Dim kolv As Double
Do While kolp < 3
n = InputBox("Введите число")
If n > 0 Then
kolp = kolp + 1
Else
kolp = 0
End If
kolv = kolv + 1
Loop
MsgBox ("Всего введено " & kolv & " чисел")
End Sub

```

Контрольные вопросы

- 1 В каких случаях используются циклы DO?
- 2 Назовите виды циклов DO. Чем они отличаются?

6 Лабораторная работа № 14. Алгоритмический язык VBA

Цель работы: изучение функций обработки строк.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить и представить отчет в виде файла в формате документа Word.

Отчет должен содержать название работы, цель работы, задание, схему алгоритма, код программы.

Основные теоретические положения

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

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.

В VBA имеются следующие функции обработки строковых выражений, представленные в таблице 3.

Таблица 3 – Функции обработки строковых выражений

Функция	Возвращаемое выражение
Asc	Возвращает ASCII-код начальной буквы строки. Синтаксис: Asc(Строка)
Функция	Возвращаемое выражение
Chr	Преобразует ASCII-код в строку. Синтаксис: Chr(Код). Например, Chr (13) – переход на новую строку
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: RTrimf(Строка)
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,]string1, string2[, compare]). Аргументы: start – числовое выражение, задающее позицию, с которой начинается каждый поиск. Если этот аргумент опущен, поиск начинается с первого символа строки, string1 – 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)

7 Лабораторная работа № 15. Алгоритмический язык VBA

Цель работы: разработка программ с использованием подпрограмм и функций.

Порядок выполнения работы

1 Ознакомиться с основными теоретическими положениями.

2 Выполнить задание, полученное у преподавателя.

3 Оформить и представить отчет в виде файла в формате документа Word.

Отчет должен содержать название работы, цель работы, задание, схему алгоритма, код программы.

Основные теоретические положения

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 и 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,
```

где 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 | ByVal] <Имя переменной> [As <Тип>],
```

где 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 Function содержит 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 список 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, т. к. 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.

Контрольные вопросы

- 1 Что такое подпрограмма?
- 2 Чем отличаются подпрограмма-функция и подпрограмма-процедура?
- 3 Как происходит передача параметров в подпрограмму?

8 Лабораторная работа № 16. Алгоритмический язык VBA

Цель работы: создание пользовательских форм; создание и использование элементов управления; ознакомление с их свойствами и методами.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание лабораторной работы № 9 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. Выполнить проверку на ввод нечисловых значений в текстовые поля с использованием функции IsNumeric.

- 3 Оформить и представить отчет в виде файла в формате документа Word. Отчет должен содержать название работы, цель работы, задание, код программы.

Основные теоретические положения

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.

Используя панель инструментов из незаполненной формы, можно сконструировать любое требуемое для приложения диалоговое окно, перетаскивая 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.

Контрольные вопросы

- 1 Назовите основные элементы управления формы.
- 2 Назовите основные свойства элементов управления формы.

9 Лабораторная работа № 17. Алгоритмический язык VBA

Цель работы: изучение процедуры обработки ошибок и использование их при решении задач.

Порядок выполнения работы

- 1 Ознакомиться с основными теоретическими положениями.
- 2 Выполнить задание, полученное у преподавателя.
- 3 Оформить и представить отчет в виде файла в формате документа Word. Отчет должен содержать название работы, цель работы, задание, схему алгоритма, код программы.

Основные теоретические положения

Отладка программ – это проверка и внесение исправлений в программу при ее разработке.

В процессе отладки программы возможны следующие виды ошибок.

Ошибки компиляции, возникающие при неправильном использовании синтаксиса инструкций, свойств и методов объектов. Например, при некорректном вводе числа скобок, неправильном имени, неполном вводе инструкции и т. д. Некоторые из этих ошибок обнаруживаются VBA при завершении набора строки с инструкцией в редакторе кода и после нажатия клавиши Enter. Строка, в которой содержится ошибка, выделяется красным цветом, и на экране отображается

диалоговое окно с сообщением о возможной причине, вызвавшей ошибку. Эти ошибки выявляются на уровне компиляции и легко исправляются;

Ошибки выполнения, возникают после успешной компиляции программы при ее выполнении. Причиной таких ошибок может быть отсутствие данных, неправильные данные, введенные пользователем (строка вместо числа, точка вместо запятой или наоборот и т. д.), некорректность вычислений (деление на ноль), некорректная информация при считывании диска и т. д. В этих и подобных случаях на экране отображается диалоговое окно с сообщением о номере ошибки и возможной причине, ее вызвавшей.

Логические ошибки, являющиеся причиной неправильной работы программы. Для нахождения таких ошибок предназначены средства отладки, которые позволяют обнаружить логические ошибки и ошибки периода выполнения, а также наблюдать за выполнением программы.

При составлении приложений важно предусмотреть, чтобы программа анализировала возможные ошибки, возникающие при ее выполнении по вине пользователя, и информировала его об этом, подсказывая, что конкретно он сделал неправильно.

Обработка ошибок – это задание реакции на ошибки, которые возникают во время выполнения программы. Целесообразно в программе создать подпрограмму – обработчик ошибок, которые могут возникнуть в данной программе на этапе выполнения.

Функции проверки типов проверяют, является ли переменная выражением специфицированного типа. Возвращают значение True, если переменная имеет заданный тип, и False – в противном случае. Функции проверки типов представлены в таблице 4.

Таблица 4 – Функции проверки типов

Функция	Проверяет
IsArray(переменная)	Является ли переменная массивом
IsDate(переменная)	Является ли переменная датой
IsEmpty(переменная)	Была ли переменная описана инструкцией Dim
IsError(переменная)	Является ли переменная кодом ошибки
IsNull(переменная)	Является ли переменная пустым значением (Null)
IsNumeric(переменная)	Является ли переменная числовым значением
IsObject(переменная)	Является ли переменная объектом

Контрольные вопросы

- 1 Какие виды ошибок существуют?
- 2 Назовите функции проверки типов.

10 Лабораторная работа № 18. Использование пакета Mathcad для решения математических задач

Цель работы: получение навыков работы в программе MathCad; получение навыков ввода данных, вычисления значений выражений и функций.

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

Основное окно приложения имеет ту же структуру, что и большинство приложений Windows. Сверху вниз располагаются заголовок окна, строка меню, панели инструментов (стандартная и форматирования) и рабочий лист, или рабочая область, документа. При запуске отображаются панели инструментов Стандартная, Форматирование и Математика.

Панель Математика предназначена для вызова на экран еще девяти панелей, с помощью которых происходит вставка математических операций в документы. Чтобы вызвать какую-либо из них, нужно нажать соответствующую кнопку. В окне редактирования формируется документ Mathcad. Новый документ получает имя Untitled (без названия) и порядковый номер. Одновременно открыто может быть до восьми документов.

Документ состоит из трех видов областей: формульных, текстовых и графических. Расположение нетекстовых блоков в документе имеет принципиальное значение. Области просматриваются системой, интерпретируются и исполняются. Просмотр идет слева направо и сверху вниз.

Для ввода текстового комментария нужно выполнить команду Text Region (Текстовая область) из пункта меню Insert или нажать клавишу с двойной кавычкой (“”), или нажать на кнопку текста на панели инструментов. Текстовая область служит для размещения текста между формулами и графиками. При этом в месте ввода появляется курсор в виде вертикального штриха, на место которого вводятся символы текста. Внутри текста курсор перемещается клавишами перемещения курсора. Переход на новую строку производится нажатием на клавишу Enter. Для окончания ввода нужно щелкнуть мышью вне текстовой области.

Для ввода формулы нужно установить указатель мыши в свободном месте окна редактирования и щелкнуть левой кнопкой мыши. Появится визир в виде красного крестика. Он указывает место, с которого начинается набор формулы.

Константами называются поименованные объекты, хранящие некоторые значения, которые не могут быть изменены. В Mathcad содержится особый вид констант – размерные, которые помимо своего числового значения характеризуются еще и указанием физической величины, к которой они относятся. Для этого указания используется символ умножения.

Переменные являются поименованными объектами, которым присвоено некоторое значение, которое может изменяться по ходу выполнения программы. Тип переменной определяется ее значением; переменные могут быть числовыми, строковыми, символьными и т. д.

Имена констант, переменных и иных объектов называют идентификаторами. Mathcad различает в идентификаторах символы верхнего и нижнего регистров. Идентификаторы Mathcad должны начинаться с буквы и могут содержать следующие символы: латинские буквы любого регистра; арабские цифры от 0 до 9; символ подчеркивания, символ процента, буквы греческого алфавита (набираются с использованием клавиши Ctrl или применяется палитра греческих букв).

Переменные должны быть предварительно определены пользователем, т. е. им необходимо хотя бы однажды присвоить значение. В качестве оператора присваивания используется знак :=, тогда как знак = отведен для вывода значения константы или переменной. Попытка использовать неопределенную переменную ведет к выводу сообщения об ошибке.

Ранжированная переменная – переменная, которая принимает ряд значений при каждом ее использовании. Для определения ранжированной переменной общего вида используется выражение:

Идентификатор:=начальное_значение,начальное_значение+шаг.. конечное_значение

Если шаг равен 1, тогда ранжированную переменную можно задавать следующим образом:

Идентификатор:=начальное_значение.. конечное_значение

Любое выражение с ранжированными переменными после знака равенства (=) создает таблицу вывода.

Функция – выражение, согласно которому проводятся некоторые вычисления с его аргументами и определяется его числовое значение. Функции в пакете Mathcad могут быть встроенные и определенные пользователем. В Mathcad имеется множество встроенных функций. Для их ввода используется команда меню Вставка – Функция или кнопка на панели инструментов.

Функция пользователя вначале должна быть определена, а затем к ней может быть произведено обращение. Функция пользователя определяется следующим образом. Задается имя функции, в скобках указывается список аргументов функции – перечень используемых в выражении переменных, разделяемых запятыми. Затем записывается знак присваивания, справа от которого записывается выражение, содержащее доступные операторы и функции с операндами и аргументами, указанными в списке аргументов.

Обращение к функции осуществляется по ее имени с подстановкой на место аргументов констант, переменных, определенных до обращения к функции, и выражений.

Операторы – элементы языка, с помощью которых можно создавать математические выражения. К ним, например, относятся символы арифметических и логических операций, знаки вычисления сумм, произведений, производной и интеграла и т. д. Операторы, обозначающие основные арифметические действия, вводятся с панели Calculator.

Контрольные вопросы

- 1 Что такое переменная?
- 2 Что такое ранжированная переменная?
- 3 Что такое функция?

11 Лабораторная работа № 19. Использование пакета Mathcad для решения математических задач

Цель работы: получение навыков построения графиков.

Основные теоретические положения

В Mathcad встроено несколько различных типов графиков, которые можно разбить на две большие группы: двухмерные и трехмерные.

Деление графиков на типы несколько условно, т. к., управляя установками многочисленных параметров, можно создавать комбинации типов графиков, а также новые типы (например, двумерная гистограмма распределения является разновидностью простого X–Y графика).

Для построения графиков используются шаблоны. Их перечень содержится в команде меню Вставка – Графики. Большинство параметров графического процессора, необходимых для построения графиков, по умолчанию задается автоматически. Поэтому для начального построения того или иного вида достаточно задать тип графика. В подменю Graph содержится список из семи основных типов графиков.

Графики любого вида, как любые объекты документа, можно выделять, заносить в буфер обмена, вызывать их оттуда и переносить в любое новое место документа. Их можно и просто перетаскивать с места на место курсором мыши, а также растягивать по горизонтали, по вертикали и по диагонали, цепляясь за специальные маркеры выделенных графиков курсором мыши.

Порядок действий при построении всех графиков одинаков. После выбора шаблона построения графика в рабочем документе открывается поле построения графика с помеченными для ввода позициями, которые нужно заполнить для определения графика.

Можно построить несколько кривых на одном и том же поле графика, для этого необходимо ввести первое выражение по оси ординат, сопровождаемое запятой. После чего непосредственно под первым выражением появится пустое поле, в которое необходимо ввести второе выражение, сопровождаемое другой запятой, чтобы получить пустое поле, и т. д.

Чтобы построить несколько независимых кривых на одном поле графика, введите два или более выражения, отделяемых запятыми по оси абсцисс, и то же самое выражение по оси ординат. Mathcad согласует выражения попарно – первое выражение по оси абсцисс с первым выражением по оси ординат, второе со

вторым и т. д. Затем рисуется график каждой пары. Можно построить до шестнадцати функций по оси ординат в зависимости от одного аргумента по оси абсцисс. Однако если для каждой кривой используется свой аргумент, то можно отобразить только до десяти графиков. Точно так же можно построить несколько графиков на одном и том же чертеже в полярных координатах, используя эту же технологию заполнения шаблона графика.

Mathcad представляет пользователю разнообразные средства форматирования графика: изменение толщины и цвета линий, вида осей координат, координатные сетки, текстовые комментарии и др.

Чтобы изменить формат графика, необходимо дважды щелкнуть мышью в области графика. Если график строится в декартовой системе координат, то появится диалоговое окно для форматирования графика. Форматирование оси графика можно также произвести, выполнив на ней двойной щелчок.

Порядок выполнения работы

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

Контрольные вопросы

- 1 Какие виды графиков существуют в Mathcad?
- 2 Как форматировать график?
- 3 Как построить несколько независимых кривых на одном поле графика?

Список литературы

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