

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

ИНФОРМАТИКА

*Методические рекомендации к лабораторным работам
для студентов специальности
1-36 01 01 «Технология машиностроения»
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»
«8» декабря 2023 г., протокол № 5

Составитель ст. преподаватель Т. Л. Шебан

Рецензент канд. техн. наук, доц. В. В. Кутузов

Методические рекомендации предназначены для студентов специальности 1-36 01 01 «Технология машиностроения» очной и заочной форм обучения.

Учебное издание

ИНФОРМАТИКА

Часть 2

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

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 21 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, 2023

Содержание

1 Лабораторная работа № 9. Алгоритмический язык VBA	4
2 Лабораторная работа № 10. Алгоритмический язык VBA	6
3 Лабораторная работа № 11. Алгоритмический язык VBA	7
4 Лабораторная работа № 12. Алгоритмический язык VBA	8
5 Лабораторная работа № 13. Алгоритмический язык VBA	10
6 Лабораторная работа № 14. Алгоритмический язык VBA	15
7 Лабораторная работа № 15. Алгоритмический язык VBA	17
8 Лабораторная работа № 16. Алгоритмический язык VBA	20
9 Лабораторная работа № 17. Алгоритмический язык VBA	22
10 Лабораторная работа № 18. Алгоритмический язык VBA	27
11 Лабораторная работа № 19. Алгоритмический язык VBA	28
12 Лабораторная работа № 20. Алгоритмический язык VBA	30
13 Лабораторная работа № 21. Алгоритмический язык VBA	33
14 Лабораторная работа № 22. Алгоритмический язык VBA	35
15 Лабораторная работа № 23. Использование пакета Mathcad для решения математических задач	40
16 Лабораторная работа № 24. Использование пакета Mathcad для решения математических задач	43
17 Лабораторная работа № 25. Использование пакета Mathcad для решения математических задач	45
Список литературы	47

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.

Задание 1

Написать линейную программу для вычисления значения выражения по формулам из таблицы 1 в соответствии с вариантом (предполагается, что значениями переменных могут быть любые действительные числа). Предусмотреть ввод и вывод данных двумя способами: с листа Excel и с помощью встроенных диалоговых окон.

Таблица 1 – Варианты заданий

Вариант	Выражение	Вариант	Выражение
1	$y = 2 \sin(x) \cos(x)$	7	$y = 2 \sin(x) \cos(x)$
2	$y = 2 \sin(x) + \cos(x)$	8	$y = \sqrt{\sin(x) \cos(x)}$
3	$y = -3 \sin^2(x)$	9	$y = \sqrt[3]{2 \sin(x) \cos(x)}$
4	$y = \sin(2x) \cos^3(x)$	10	$y = 2 \sin(x) + \cos(x)$
5	$y = \frac{2 \sin(x)}{\cos(x^2)}$	11	$y = x^2 + 4x - 5$
6	$y = 2 \tan(x + 1)$	12	$y = \tan^3(x) + \sqrt{\cos(x)}$

Контрольные вопросы

- 1 Назовите способы ввода и вывода данных.
- 2 Назовите основные типы данных VBA.
- 3 Какие кнопки по умолчанию размещены на окне вывода?

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
```

Задание 2

Написать программу для решения задачи из таблицы 2 в соответствии с вариантом.

Таблица 2 – Варианты заданий

Вариант	Задача
1	Составить программу, которая проверяет, являются ли введенные три числа углами треугольника
2	Составить программу, которая запрашивает два числа и ответ на произведение этих чисел, проверяет его и выводит сообщение «Правильно» или «Вы ошиблись» и правильный результат
3	Составить программу для нахождения минимального из трех чисел
4	Составить программу, которая запрашивает два числа и ответ на сумму этих чисел, проверяет его и выводит сообщение: «Правильно» или «Вы ошиблись» и правильный ответ
5	Составить программу для нахождения максимального из трех чисел
6	Ввести число и определить, делится ли оно на 3 и на 7 одновременно
7	Ввести число и определить, делится ли оно на 2, 3 и 5 одновременно
8	Ввести три числа и определить, есть ли среди них 0
9	Ввести число и определить, является ли оно целым числом

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
```

Задание 3

Написать программу с использованием оператора цикла с параметром For Next в соответствии с вариантом из таблицы 3.

Таблица 3 – Варианты заданий

Вариант	Задача
1	Найти количество положительных чисел в последовательности из n чисел
2	Найти сумму положительных чисел в последовательности из n чисел
3	Найти количество нулей в последовательности из n чисел
4	Вводится n чисел. Найти их среднее арифметическое
5	Найти факториал числа (факториалом числа n называют произведение целых чисел от 1 до n : $n! = 1 \times 2 \times \dots \times n$)
6	Вывести таблицу соответствий между весом в фунтах и килограммах для значений от n фунтов до m фунтов, n и m вводятся. 1 фунт = 400 г

Окончание таблицы 3

Вариант	Задача
7	Ввести n целых чисел и определить, сколько из них попадают в диапазон – $m < \text{число} < t$, n вводится и должно быть больше 0
8	Возвести число m в целую степень n , не используя операции возведения в степень
9	Ввести n вещественных чисел и определить, что больше: их сумма или произведение

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

```

Задание 4

Написать алгоритм и программу обработки одномерного массива согласно варианту.

Вариант 1. Создать одномерный массив из N целочисленных значений. Найти наибольший среди положительных элементов массива и вывести его на экран.

Вариант 2. Создать одномерный целочисленный массив из N значений. Подсчитать среднее арифметическое значение всех элементов и вывести его на экран.

Вариант 3. Создать одномерный целочисленный массив из N значений. Расставить в нем элементы в обратном порядке.

Вариант 4. Создать одномерный целочисленный массив из N значений. Поменять местами каждую пару чисел, например, A_1 с A_2 , A_3 с A_4 и т. д.

Вариант 5. Создать одномерный массив из N целочисленных значений. Вывести все отрицательные элементы массива и подсчитать их количество.

Вариант 6. Создать одномерный массив из N целочисленных значений. Вывести все положительные элементы массива и подсчитать их количество.

Вариант 7. Создать два одномерных массива по 10 целочисленных значений каждый. Сформировать третий массив из 20 элементов следующим образом: на нечетные места ставить элементы из первого массива, на четные места – из второго.

Вариант 8. Создать два одномерных массива по 10 целочисленных значений каждый. Сформировать третий массив из 20 элементов следующим образом: сначала ставить элементы из первого массива, затем – из второго.

Вариант 9. Создавать одномерный массив из 20 целочисленных значений. Вывести элементы, стоящие на нечетных местах.

Вариант 10. Создать два одномерных массива по 20 целочисленных значений. Совершить обмен данными: в одном массиве – с 1-го по 10-й элемент, в другом – с 11-го по 20-й элемент, т. е. X_1 на Y_{11} , Y_{11} на X_1 и т. д.

Вариант 11. Создать одномерный массив из N целочисленных значений. Расставить в нем в том же порядке сначала положительные элементы, затем отрицательные, затем нулевые.

Вариант 12. Создать одномерный массив из N целочисленных значений. Найти минимальное среди них, вывести его и индекс данного элемента на экран.

Вариант 13. Создать одномерный массив из N целочисленных значений. Найти минимальное и максимальное значения, вывести их разность на экран.

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 4.

Таблица 4 – 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)

Операторы циклов с постусловием: Do...Loop While, Do...Loop Until. Синтаксис операторов циклов с постусловием представлен в таблице 5.

Таблица 5 – 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)

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.

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
```

Задание 5

Написать программу с использованием оператора повтора с предусловием или постусловием в соответствии с вариантом.

Вариант 1. Вводить последовательность до тех пор, пока не встретятся три подряд идущих положительных числа, прервать ввод и сообщить, сколько во введенной последовательности было всего чисел.

Вариант 2. Вводить последовательность до тех пор, пока не встретятся три

поряд идущих положительных числа, прервать ввод и сообщить, сколько во введенной последовательности было положительных чисел.

Вариант 3. Вводить последовательность до тех пор, пока не встретятся три подряд идущих положительных числа, прервать ввод и сообщить, сколько во введенной последовательности было отрицательных чисел.

Вариант 4. Определить сколько натуральных подряд идущих четных чисел нужно сложить (найти минимальное число таких слагаемых), чтобы их сумма была больше введенного числа.

Вариант 5. Последовательно вводятся числа до тех пор, пока во введенной совокупности не окажется три нуля. Вывести количество введенных чисел.

Вариант 6. Дано натуральное число. Определить номер цифры 3 в нем, считая от конца числа. Если таких цифр в числе несколько – должен быть определен номер самой правой из них.

Вариант 7. Дано натуральное число. Если в нем есть цифры «2» и «5», то определить, какая из них расположена в числе левее. Если одна или обе эти цифры встречаются в числе несколько раз, то должны быть рассмотрены самые левые из одинаковых цифр.

Вариант 8. Дано натуральное N и первый член бесконечного ряда: $Y_1 = 1$. Вычислить сумму членов бесконечного ряда, образованного по следующему рекуррентному соотношению: $Y_i = 2 \times Y_{i-1}$ (т. е. $S = 1 + 2 + 4 + 8 + 16 + \dots$). Вычисление суммы продолжать до тех пор, пока соблюдается условие $|Y_i - Y_{i-1}| < N$.

Вариант 9. Дано натуральное число. Верно ли, что сумма его цифр меньше числа A , введенного с клавиатуры?

Вариант 10. Дано натуральное число. Верно ли, что произведение его цифр больше числа B , введенного с клавиатуры?

Вариант 11. Дано натуральное число. Определить количество и сумму его цифр.

Вариант 12. Дано натуральное число. Определить среднее арифметическое и произведение его цифр.

Пример выполнения задания

Составим алгоритм следующей задачи, используя различные циклы с предусловием и постусловием: *вводите числа, пока их сумма не превысит введенного числа m .*

1 Решение с использованием цикла с предусловием **Do While ...Loop**.

```
Public Sub uuu()
  Dim x As Integer
  Dim m As Integer
  Dim s As Integer
  Dim i As Integer
  m = InputBox("Введите число")
  MsgBox ("Вводите числа")
  i = 1
  s = InputBox("Введите 1 число")
  Do While s <= m
    i = i + 1
```

```

x = InputBox("Введите " & i & "число")
s = s + x
Loop
MsgBox ("Количество введенных чисел " & i)
End Sub

```

2 С использованием цикла **Do Until ... Loop**.

```

Public Sub uuu()
Dim x As Integer
Dim m As Integer
Dim s As Integer
Dim i As Integer
m = InputBox("Введите число")
MsgBox ("Вводите числа")
i = 1
s = InputBox("Введите 1 число")
Do Until s > m
i = i + 1
x = InputBox("Введите " & i & "число")
s = s + x
Loop
MsgBox ("Количество введенных чисел " & i)
End Sub

```

3 С использованием цикла **Do ...Loop While**.

```

Public Sub uuu()
Dim x As Integer
Dim m As Integer
Dim s As Integer
Dim i As Integer
m = InputBox("Введите число")
MsgBox ("Вводите числа")
i = 0
s = 0
Do
i = i + 1
x = InputBox("Введите " & i & "число")
s = s + x
Loop While s <= m
MsgBox ("Количество введенных чисел " & i)
End Sub

```

4 С использованием цикла **Do ...Loop Until**.

```

Public Sub uuu()
Dim x As Integer
Dim m As Integer
Dim s As Integer
Dim i As Integer
m = InputBox("Введите число")
MsgBox ("Вводите числа")

```

```

i = 0
s = 0
Do
  i = i + 1
  x = InputBox("Введите " & i & "число")
  s = s + x
Loop Until s > m
MsgBox ("Количество введенных чисел " & i)
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 6.

Задание 6

Выполнить задание согласно варианту, используя функции обработки строк.

Вариант 1. Заменить в тексте букву «о» на цифру «0», а букву «з» на цифру «3». Вывести количество этих замен.

Вариант 2. Определить, является ли введенное 6-значное число числом-палиндромом. Числа-палиндромы – это такие числа, которые читаются одинаково как справа налево, так и слева направо, т. е. 654456.

Вариант 3. Задан текст. Проверить, какая из букв «а» или «е» встречается чаще.

Таблица 6 – Функции обработки строковых выражений

Функция	Возвращаемое выражение
Asc	Возвращает ASCII-код начальной буквы строки. Синтаксис: Asc(Строка)
Chr	Преобразует ASCII-код в строку. Синтаксис: Chr(Код). Например, Chr(13) – переход на новую строку
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: RTrim(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)

Вариант 4. Задан текст. Распечатать номера позиций, на которых встречается буква «к».

Вариант 5. Определить, сколько в тексте слов заканчивается на букву «о».

Вариант 6. Известно, что в заданный текст входит хотя бы одна буква «а», причем не на последнем месте. Напечатать по одной литере текста, следующей непосредственно за буквой «а».

Вариант 7. Определить, сколько раз в тексте встречается введенное слово.

Вариант 8. Определить, какой процент слов в тексте содержит хотя бы одну букву «d».

Вариант 9. Напечатать те слова из текста, в которых нет удвоенной буквы.

Вариант 10. Удалить из текста все удвоенные буквы.

Вариант 11. Из заданного текста удалить последовательность символов между двумя двоеточиями (хотя бы одна пара двоеточий имеется).

Вариант 12. Перед каждой цифрой в тексте вставить знак «+».

Вариант 13. Напечатать самое длинное слово из заданного текста.

Вариант 14. Напечатать удвоенные буквы в строке и их позиции.

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

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

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 ByVal) 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 ByVal), то при вызове подпрограммы передается сама переменная, а не ее копия. Изменение параметра-переменной приводит к изменению самого фактического параметра в вызывающей подпрограмме.

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

Задание 7

Решить задачу по вариантам. Задание выполнить двумя способами: используя подпрограмму-процедуру и подпрограмму-функцию.

Вариант 1. Рассчитать значение $x = \frac{\sqrt{6+6}}{2} + \frac{\sqrt{13+13}}{2} + \frac{\sqrt{21+21}}{2}$.

Вариант 2. Даны два натуральных числа. Выяснить, в каком из них сумма цифр больше.

Вариант 3. Рассчитать значение $x = \frac{\sqrt{6+17}}{6+\sqrt{17}} + \frac{\sqrt{17+13}}{17+\sqrt{13}} + \frac{\sqrt{21+32}}{21+\sqrt{32}}$.

Вариант 4. Даны основания и высоты двух равнобедренных трапеций. Найти сумму их периметров и сумму их площадей.

Вариант 5. Найти значение выражения $x = \frac{2 \cdot 5! + 6 \cdot 8!}{6! + 4!}$, где $n!$ означает факториал числа n .

Вариант 6. Рассчитать значение $x = \frac{\sqrt{6+1}}{1+\sqrt{8}} + \frac{\sqrt{10+1}}{1+\sqrt{12}} + \frac{\sqrt{14+1}}{1+\sqrt{16}}$.

Вариант 7. Даны стороны двух треугольников. Найти сумму их периметров и сумму их площадей.

Вариант 8. Написать программу вычисления суммы факториалов всех нечетных чисел от 1 до 9.

Вариант 9. Даны три квадратных уравнения $ax^2 + bx + c = 0$, $bx^2 + ax + c = 0$, $cx^2 + ax + b = 0$. Определить сколько из них имеют вещественные корни.

Вариант 10. Даны 6 различных чисел. Определить минимальное из них. (определить процедуру, находящую минимум из двух различных чисел).

Вариант 11. Даны 6 различных чисел. Определить максимальное из них. (определить процедуру, находящую максимум из двух различных чисел).

Вариант 12. Дано n окружностей с радиусами r_1, r_2, \dots, r_n соответственно. Найти сторону прямоугольника, площадь которого равна сумме n площадей окружностей, если одна сторона прямоугольника равна a . Вычисление площади круга оформить в виде процедуры.

Вариант 13. Найти значение выражения $x = \frac{7 \cdot 8! + 9 \cdot 10!}{11! + 12!}$, где $n!$ означает факториал числа n .

Вариант 14. Написать программу вычисления суммы факториалов всех четных чисел от 2 до 10.

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 лабораторных работ № 9 и 10 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 активизируется командой View – Property Window.

Задание 8

Создать пользовательскую форму для вычисления по вариантам, предусмотрев на ней функцию очистки окон.

Вариант 1. Вычислить $\gamma = |x - y| - (1 + 2x)^y + e^{\sqrt{|x-y|}}$.

Вариант 2. Вычислить $f(x, y, z) = \frac{x+y+z}{x^2+y^2+z^2}$.

Вариант 3. Вычислить площадь прямоугольного треугольника по двум катетам.

Вариант 4. Вычислить $f(x, y) = \frac{x}{1+y} + \frac{y}{1+x} + \frac{y}{y+x}$.

Вариант 5. Вычислить площадь куба по его стороне.

Вариант 6. Вычислить $f(x, y, z) = \frac{x+y+z}{x \cdot y \cdot z}$.

Вариант 7. $f(x, y, z) = \frac{xy}{z} + \frac{yz}{x} + \frac{xz}{y}$.

Вариант 8. Вычислить дискриминант квадратного уравнения по заданным коэффициентам.

Вариант 9. Вычислить $f(x, y) = (x + y)(x^2 + y^2)(x^3 + y^3)$.

Вариант 10. Вычислить объем шара по заданному радиусу.

Вариант 11. Вычислить объем цилиндра по заданному радиусу и высоте.

Вариант 12. Вычислить $f(x, y, z) = \frac{x \cdot y \cdot z}{x + y^2 + z^3}$.

Вариант 13. Вычислить $f(x, y, z) = \frac{x + 2y + 3z}{x^3 + y^2 + z}$.

Вариант 14. Вычислить произведение четырех вещественных чисел.

Вариант 15. Вычислить длину гипотенузы по заданным длина катетов.

Контрольные вопросы

1 Назовите основные элементы управления формы.

2 Назовите основные свойства элементов управления формы.

9 Лабораторная работа № 17. Алгоритмический язык VBA

Цель работы: освоение основных приемов работы с текстовыми данными и файлами.

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

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

FSO предоставляет разработчику возможность работы с файлами в ясной объектно-ориентированной манере. FSO опирается на библиотеку Microsoft Scripting Runtime Library (файл Scrrun.dll), которая совместно используется несколькими средами разработки и позволяет, в частности, расширить функции написания сценариев.

Для использования этой библиотеки надо установить ссылку на Microsoft Scripting Runtime в диалоговом окне References – VBAProject, которое отображается на экране выбором команды Tools – References. После этого можно приступить к программированию объектов, функционирование которых обеспечивается данной библиотекой.

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

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

Синтаксис: `GetFile (filespec)`.

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

Синтаксис: `FileExists(folderspec)`.

Метод `FileExists` возвращает `True`, если диск существует, и `False` – в противном случае.

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

Синтаксис: `CreateTextFile(filename [, overwrite [, unicode]])`,
где `filename` – строковое выражение, идентифицирующее создаваемый текстовый файл;

`overwrite` – параметр, принимающий значение равно `True`, если указанный файл уже существует и его следует перезаписать, и `False`, если существующий файл не перезаписывается;

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

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

`Delete` – удаляет файл, `Copy` – копирует файл, `Move` – перемещает файл, `OpenAsTextStream` – создает объект `Textstream` и специфицирует имя открываемого текстового файла. Этот объект используется для чтения, записи и добавления данных в файл:

`OpenAsTextStream ([iomode, [format]])`,

где `iomode` – задает тип операции, производимой над файлом. Допустимые значения:

`ForReading` или 1 – для чтения;

`ForWriting` или 2 – для записи;

`ForAppending` или 8 – для добавления данных.

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

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

где `filename` – строковое выражение, идентифицирующее открываемый текстовый файл;

`iomode` – задает тип операции, производимой над файлом. Допустимые значения:

`ForReading` или 1 – для чтения;

`ForAppending` или 8 – для добавления данных;

`create` – логический параметр, который указывает, надо ли создавать новый файл, если файл с данным именем не существует;

`format` – задает тип файла.

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

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

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

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

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

Метод `SkipLine` пропускает одну строку при считывании данных из файла.

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

Метод `Close` закрывает открытый `Textstream`. По завершении работы с файлом никогда не забывайте его закрывать.

Свойство `Line`, `Column` возвращают номер строки и столбца, в котором находится текущий символ.

Свойство `AtEndOfStream` возвращает `True`, если указатель достиг конца файла.

Свойство `AtEndOfLine` возвращает `True`, если указатель достиг конца строки.

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

```
Sub MultTable()
    Dim fso As FileSystemObject
    Dim ts As TextStream
    Dim i as Integer, j as Integer
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set ts = fso.OpenTextFile("C:\Table.txt", ForWriting, True)
    For i = 1 To 9
        For j = 1 To 9
            ts.Write (Format(i * j, "00"))
            ts.Write (Space(1))
        Next j
        ts.WriteLine ' Переход на новую строку
    Next i
    ts.Close
End Sub
```

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


```

Sub ReadIt()
    Dim fso As FileSystemObject
    Dim ts As TextStream
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set ts = fso.OpenTextFile("C:\Table.txt", ForReading)
    Do While Not ts.AtEndOfStream
        Debug.Print ts.ReadLine
    Loop
    ts.Close
End Sub

```

Задание 9

Выполнить задания согласно варианту, используя переключатели.

Вариант 1.

1 Выбирается число от 1 до 4, определяющее пору года. Вывести название этой поры.

2 На трех заводах «Альфа», «Плутон» и «Рубин» иногда происходят аварии. Разместить сведения о количестве аварий за каждый год (взять последние 4 года) на рабочем листе. Для выбранного пользователем завода и года вывести число аварий и их процент, принимая за 100 % общее число аварий за текущий год.

Вариант 2.

1 Выбирается число от 1 до 7, определяющее день недели. Дать название этого дня.

2 Известен состав из четырех учредителей (акционеров) АО «Рога и копыта». Оформить лист, где будут указаны сведения о составе учредителей: ФИО учредителя, количество обыкновенных акций, количество привилегированных акций. Для выбранного учредителя и вида акции вывести общую стоимость акций данного вида, а также доли акционера (в процентах) от общей стоимости всех акций. Стоимости одной акции каждого вида известны и должны задаваться в отдельных ячейках

Вариант 3.

1 Выбирается число от 1 до 5. Дать название этого числа.

2 Подготовить лист, где будут приведены цены для пяти наименований товаров. Для выбранного товара вывести цену и цену со скидкой. Величина скидки (в процентах) задается.

Вариант 4.

1 Вводится нецелое число. Вывести либо его целую часть, либо дробную в зависимости от выбора пользователя.

2 Разместить на листе сведения о заработной плате работников некоторого малого предприятия (пять работников): ФИО работника, оклад. Для выбранного работника рассчитать подоходный налог (9 % от оклада за вычетом минимального размера оплаты труда), пенсионный налог (1 % от оклада), профсоюзный взнос (1 % от оклада), сумму к выдаче. Так как минимальный размер оплаты труда периодически меняется – вводить его в поле ввода.

Вариант 5.

1 Банк предлагает три вида срочных вкладов: на 3 месяца под 27 %, на 6 месяцев под 29 % и на год под 30 %. Вкладчик положил N у. е. на один из срочных вкладов. Какую сумму он получит по истечении срока?

2 В таблице приведены сведения о количестве легковых автомобилей, выпущенных ведущими мировыми производителями в первом полугодии текущего года по месяцам. Для выбранного пользователем производителя и месяца вывести число автомобилей, произведенных за этот месяц, а также общее число автомобилей, произведенных им за полугодие и их процент, принимая за 100 % общее число автомобилей.

Вариант 6.

1 Задано расстояние в метрах. Пересчитать это расстояние в километрах, милях, футах или ярдах на выбор пользователя (1 миля = 1,609 км, 1 м = 1,094 ярда, 1 м = 3,281 фута).

2 Подготовить расчет стоимости автомобиля в зависимости от его дополнительной комплектации.

Вариант 7.

1 Дан объем в литрах. Пересчитать этот объем в пинтах, галлонах, бушелях и квартах (английские меры объема жидких и сыпучих тел) на выбор пользователя (1 л = 1,706 пинты, 1 л = 0,220 галлона, 1 бушель = 36,35 л, 1 кварта = 1,136 л).

2 Подготовить пользовательскую форму для расчета оплаты услуг в зависимости от объема выполненной работы.

Вариант 8.

1 Дана масса в килограммах. Пересчитать эту массу в пудах, фунтах, центнерах или тоннах на выбор пользователя (1 пуд = 16,38 кг, 1 фунт = 0,409 кг, 1 т = 1000 кг, 1 ц = 100 кг).

2 Подготовить расчет стоимости санаторного лечения в зависимости от срока пребывания (7, 12, 15 и 24 дня) и полноты питания (завтрак; завтрак и обед; завтрак, обед и ужин).

Вариант 9.

1 Дано расстояние в метрах. Пересчитать его в верстах, саженьях, аршинах или вершках на выбор пользователя (1 верста = 1,067 км, 1 сажень = 2,134 м, 1 аршин = 0,7112 м, 1 вершок = 4,445 см).

2 Подготовить расчет стоимости ремонта квартиры в зависимости от метража и проделанной работы.

Вариант 10.

1 Разработать пользовательскую форму для нахождения периметра или площади прямоугольника по заданным сторонам.

2 Разработать программу, «расспрашивающую» покупателя авиабилета о характере заказа: пункт назначения, расположение кресла (в середине, у прохода, у окна), тип салона (для курящих или не курящих). Рассчитать стоимость билета с учетом надбавок за расположение кресла и типа салона.

Контрольные вопросы

- 1 Может ли шаг изменения переменной цикла быть отрицательным?
- 2 Можно ли досрочно выйти из цикла?

10 Лабораторная работа № 18. Алгоритмический язык VBA

Цель работы: изучение условного оператора Select Case.

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

Условный оператор Select Case используется в тех случаях, когда необходимо проверять одно и то же значение, сравнивая его с различными выражениями.

Синтаксис оператора:

```
Select Case <выражение>
Case <список выражений 1>
<операторы 1>
Case <список выражений N>
<операторы N>
[Case Else
<операторы N+1>]
End Select
```

Здесь

<выражение> – любое численное или строковое выражение;

<список выражений i> – представляют список выражений, отделенных друг от друга запятыми или (если в интервале), to in to ik;

<операторы i> - (i=1,N) – операторы, которые выполняются, если <выражение i> совпадает с любым компонентом <список выражений i>;

[Case Else

<операторы N+1>] необязательная конструкция.

Работает данный оператор следующим образом: вычисляется <выражение>; если значение <выражение> совпадает со значением из Case <список выражений i>, то выполняются соответствующие <операторы i> и затем управление передается на оператор, следующий за оператором Select Case; если значение <выражение> не совпадает ни с одним из Case <список выражений i>, то при

наличии конструкции Case Else выполняются <операторы N+1>; если конструкция Case Else отсутствует, то сразу выполняется оператор, следующий за оператором Select Case.

Задание 10

Решить задачу по вариантам. В случае ввода недопустимых данных предусмотреть вывод сообщения «Ошибка ввода данных».

Вариант 1. По запрашиваемому у пользователя числу, определить пору года: 1 – зима, 2 – весна и т. д.

Вариант 2. По запрашиваемому у пользователя номеру дня недели, вывести одно из сообщений: «Рабочий день», «Выходной».

Вариант 3. По введенному числу определить месяц весны: 3 – март,

Вариант 4. По введенному числу определить месяц осени: 9 – сентябрь,

Вариант 5. По введенному числу озвучить оценку: «5» – отлично,

Вариант 6. По введенному числу определить цвет спектра радуги (подсказка: каждый охотник желает знать где сидит фазан, т. е. «красный», «оранжевый», «желтый», ...).

Вариант 7. По введенному времени в формате *часы...минуты* определить время суток: утро, день, вечер, ночь.

Вариант 8. По введенному числу определить месяц зимы: 12 – декабрь,

Вариант 9. По введенному числу определить день недели: 1 – понедельник, 2 – вторник,

Вариант 10. По введенному числу определить месяц лета: 6 – июнь,

Вариант 11. По введенному времени в формате *часы...минуты* определить не закончен ли рабочий день, длящийся с 9.00 до 18.00.

Вариант 12. По введенному числу озвучить цифру: «1» – один

Вариант 13. По введенной дате в формате *день...месяц* определить пору года.

Контрольные вопросы

- 1 Назначение оператора Select Case.
- 2 Порядок работы оператора Select Case.

11 Лабораторная работа № 19. 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 7.

Задание 11

Для задачи из лабораторной работы № 10 предусмотреть обработку ошибок.

Таблица 7 – 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.

12 Лабораторная работа № 20. Алгоритмический язык VBA

Цель работы: изучение свойств элемента управления OptionButton; использование его для решения 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

Элемент управления OptionButton (переключатель) позволяет выбрать один из нескольких взаимоисключающих параметров или действий. Переключатели обычно отображаются группами, созданными с помощью элемента управления Frame, обеспечивая возможность выбора альтернативного варианта.

Основными событиями переключателя являются события Click и Change. Наиболее часто используемые свойства элемента управления OptionButton представлены в таблице 8.

Таблица 8 – Свойства элемента управления OptionButton

Свойство	Описание
Value	Возвращает True, если переключатель выбран и False в противном случае
Enabled	Допустимые значения: True (пользователь может выбрать переключатель) и False (в противном случае)
Visible	Допустимые значения: True (переключатель отображается во время выполнения программы) и False (в противном случае)
Caption	Надпись, отображаемая рядом с переключателем

Задание 12

Выполнить задания согласно варианту, используя переключатели.

Вариант 1.

1 Выбирается число от 1 до 4, определяющее пору года. Вывести название этой поры.

2 На трех заводах «Альфа», «Плутон» и «Рубин» иногда происходят аварии. Разместить сведения о количестве аварий за каждый год (взять последние 4 года) на рабочем листе. Для выбранного пользователем завода и года вывести число аварий и их процент, принимая за 100 % общее число аварий за текущий год.

Вариант 2.

1 Выбирается число от 1 до 7, определяющее день недели. Дать название этого дня.

2 Известен состав из 4-х учредителей (акционеров) АО «Рога и копыта». Оформить лист, где будут указаны сведения о составе учредителей: ФИО учредителя, количество обыкновенных акций, количество привилегированных акций. Для выбранного учредителя и вида акции вывести общую стоимость акций данного вида, а также доли акционера (в процентах) от общей стоимости всех акций. Стоимости одной акции каждого вида известны и должны задаваться в отдельных ячейках

Вариант 3.

1 Выбирается число от 1 до 5. Дать название этого числа.

2 Подготовить лист, где будут приведены цены для пяти наименований товаров. Для выбранного товара вывести цену и цену со скидкой. Величина скидки (в процентах) задается.

Вариант 4.

1 Вводится нецелое число. Вывести либо его целую часть, либо дробную в зависимости от выбора пользователя.

2 Разместить на листе сведения о заработной плате работников некоторого малого предприятия (пять работников): ФИО работника, оклад. Для выбранного работника рассчитать подоходный налог (9 % от оклада за вычетом минимального размера оплаты труда), пенсионный налог (1 % от оклада), профсоюзный взнос (1 % от оклада), сумму к выдаче. Так как минимальный размер оплаты труда периодически меняется – вводить его в поле ввода.

Вариант 5.

1 Банк предлагает три вида срочных вкладов: на 3 месяца под 27 %, на 6 месяцев под 29 % и на год под 30 %. Вкладчик положил N у. е. на один из срочных

вкладов. Какую сумму он получит по истечении срока?

2 В таблице приведены сведения о количестве легковых автомобилей, выпущенных ведущими мировыми производителями в первом полугодии текущего года по месяцам. Для выбранного пользователем производителя и месяца вывести число автомобилей, произведенных за этот месяц, а также общее число автомобилей, произведенных им за полугодие и их процент, принимая за 100 % общее число автомобилей.

Вариант 6.

1 Задано расстояние в метрах. Пересчитать это расстояние в километрах, милях, футах или ярдах на выбор пользователя (1 миля = 1,609 км, 1 м = 1,094 ярда, 1 м = 3,281 фута).

2 Подготовить расчет стоимости автомобиля в зависимости от его дополнительной комплектации.

Вариант 7.

1 Дан объем в литрах. Пересчитать этот объем в пинтах, галлонах, бушелях и квартах (английские меры объема жидких и сыпучих тел) на выбор пользователя (1 л = 1,706 пинты, 1 л = 0,220 галлона, 1 бушель = 36,35 л, 1 кварта = 1,136 л).

2 Подготовить пользовательскую форму для расчета оплаты услуг в зависимости от объема выполненной работы.

Вариант 8.

1 Дана масса в килограммах. Пересчитать эту массу в пудах, фунтах, центнерах или тоннах на выбор пользователя (1 пуд = 16,38 кг, 1 фунт = 0,409 кг, 1 т = 1000 кг, 1 ц = 100 кг).

2 Подготовить расчет стоимости санаторного лечения в зависимости от срока пребывания (7, 12, 15 и 24 дня) и полноты питания (завтрак; завтрак и обед; завтрак, обед и ужин).

Вариант 9.

1 Дано расстояние в метрах. Пересчитать его в верстах, саженьях, аршинах или вершках на выбор пользователя (1 верста = 1,067 км, 1 сажень = 2,134 м, 1 аршин = 0,7112 м, 1 вершок = 4,445 см).

2 подготовить расчет стоимости ремонта квартиры в зависимости от метража и проделанной работы.

Вариант 10.

1 Разработать пользовательскую форму для нахождения периметра или площади прямоугольника по заданным сторонам.

2 Разработать программу, «расспрашивающее» покупателя авиабилета о характере заказа: пункт назначения, расположение кресла (в середине, у прохода, у окна), тип салона (для курящих или не курящих). Рассчитать стоимость билета с учетом надбавок за расположение кресла и типа салона.

Пример – Разработать программу выполнения одной из четырех арифметических операций над двумя числами по выбору пользователя. Исполняемая операция устанавливается за счет выбора соответствующего переключателя.

Контрольные вопросы

- 1 Назовите основные свойства элемента управления `OptionButton`.
- 2 Назовите основные методы элемента управления `OptionButton`.

13 Лабораторная работа № 21. Алгоритмический язык VBA

Цель работы: изучение свойств, событий и методов элемента управления `ListBox`; использование списков при решении задач.

Порядок выполнения работы

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

Основные теоретические положения

Элемент управления `ListBox` применяется для хранения списка значений. Из списка пользователь может выбрать одно или несколько значений, которые в последующем будут использоваться в тексте программы.

Наиболее часто используемые свойства элемента управления `ListBox` представлены в таблице 9, а методы – в таблице 10.

Таблица 9 – Свойства элемента управления `ListBox`

Свойство	Описание
<code>ListIndex</code>	Возвращает номер текущего элемента списка. Нумерация элементов списка начинается с нуля
<code>ListCount</code>	Возвращает число элементов списка
<code>TopIndex</code>	Возвращает элемент списка с наибольшим номером
<code>ColumnCount</code>	Устанавливает число столбцов в списке
<code>TextColumn</code>	Устанавливает столбец в списке, элемент которого возвращается свойством <code>Text</code>
<code>Enabled</code>	Допустимые значения: <code>True</code> (запрещен выбор значения из списка пользователем) и <code>False</code> (в противном случае)
<code>Text</code>	Возвращает выбранный в списке элемент
<code>List</code>	Возвращает элемент списка, стоящий на пересечении указанных строки и столбца. Синтаксис <code>List (row, column)</code>
<code>RowSource</code>	Устанавливает диапазон, содержащий элементы списка
<code>ControlSource</code>	Устанавливает диапазон (ячейку), куда возвращается выбранный элемент из списка
<code>Selected</code>	Допустимые значения: <code>True</code> (если элемент списка выбран) и <code>False</code> (в противном случае). Используется для определения выделенного текста, когда свойство <code>MultiSelect</code> имеет значение <code>fmMultiSelectMulti</code> или <code>fmMultiSelectExtended</code>

Таблица 10 – Методы элемента управления ListBox

Метод	Описание
Clear	Удаляет все элементы из списка
RemoveItem	Удаляет из списка элементы с указанным номером. Синтаксис: Remove Item (index), index – номер, удаляемого из списка элемента
AddItem	Добавляет элемент в список Синтаксис: AddItem ([Item, [VarIndex]]), -item – элемент, добавляемый в список, -varindex – номер добавляемого элемента

Задание 13

Разработать программу, содержащую список.

Вариант 1. Дан одномерный массив. Отсортировать его по возрастанию элементов. Вывести в один список – исходный массив, в другой – отсортированный.

Вариант 2. Дан одномерный массив. Заменить четные числа на 1, нечетные – на -1. Вывести в один список – исходный массив, в другой – преобразованный.

Вариант 3. Дан одномерный массив. Вывести в один список – исходный массив, в другой – только элементы, кратные трем.

Вариант 4. Вычислить $\frac{x^2}{2}, \frac{x^3}{3}, \dots, \frac{x^{11}}{11}$ для указанного значения x .

Вариант 5. Вывести члены арифметической прогрессии. Значение первого члена, разность и количество членов задаются (формула n -го члена: $a_n = a_1 + d(n - 1)$).

Вариант 6. Вывести члены геометрической прогрессии. Значение первого члена, знаменатель и количество членов задаются (формула n -го члена: $b_n = b_1 q^{n-1}$).

Вариант 7. Разработать программу, содержащую многостолбцовый список. Рассчитать таблицу значений функции $y = \sqrt{x^2 + k^2}$, где x меняется от -2 до 2 с шагом 0,1, а k – параметр, задаваемый пользователем. Таблицу поместить в двухстолбцовый список.

Вариант 8. Составить таблицу перевода километров в мили на интервале от 10 до 50 с шагом 10 (1 миля = 1,609 км). Таблицу поместить в двухстолбцовый список.

Вариант 9. Составить таблицу перевода метров в ярды на интервале от 2 до 10 с шагом 0,5 (1 метр = 1,094 ярда). Таблицу поместить в двухстолбцовый список.

Вариант 10. Составить таблицу перевода метров в футы на интервале от 10 до 50 с шагом 5 (1 м = 3,281 фута). Таблицу поместить в двухстолбцовый список.

Вариант 11. Составить таблицу квадратных корней из чисел от a до b с шагом 0,1. Значения a и b задаются ($a < b$). Таблицу поместить в двухстолбцовый список.

Вариант 12. Дан массив размерность $n \times m$. Найти максимальный по модулю элемент массива. Вывести массив в список и найденный элемент в поле.

Вариант 13. Дан массив размерность $n \times m$. Найти минимальный элемент массива. Вывести массив в список и найденный элемент в поле.

Вариант 14. Дан массив размерность $n \times m$. Найти индекс минимального элемента массива. Вывести массив в список и найденный индекс в поле.

Контрольные вопросы

- 1 Назовите основные свойства элемента управления ListBox.
- 2 Назовите основные методы элемента управления ListBox.

14 Лабораторная работа № 22. Алгоритмический язык VBA

Цель работы: изучение основных свойств и методов элемента управления Image (*рисунок*); построение графика на листе Excel и на пользовательской форме.

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

В Excel различаются два вида диаграмм: внедренная на рабочий лист (объект ChartObject) и созданная на специальном листе для диаграмм (объект Charts). Свойства и методы этих объектов представлены в таблицах 11 и 12 соответственно.

Таблица 11 – Свойства объектов ChartObject и Charts

Свойство	Значение
ChartArea	Возвращает объект ChartArea — область на листе, отведенная под диаграмму. В следующем примере устанавливается цвет элементов объекта ChartArea. With Charts { "Диаграмма") .ChartArea . Interior . ColorIndex = 3 . Border . ColorIndex = 5 End With

Окончание таблицы 11

Свойство	Значение
ChartTitle	Возвращает объект ChartTitle, являющийся заголовком диаграммы. В следующем примере задается заголовок диаграммы отчет : With Charts ("Отчет") .HasTitle = True .ChartTitle .Text = "Продажи за май" End With
ChartType	Возвращает тип диаграммы: xlArea, xlBar, xlColumn, xlLine, xlPie, xlRadar, xlXYScatter, xlCombination, xl3DArea, xlSDBar, xl3DColumn, xl3DLine, xl3DPie, xlSDSurface, xlDoughnut
Legend	Возвращает объект Legend. Используется для изменения легенды. Например: ChartObjects(1).Chart.Legend.Font.Bold=True

Таблица 12 – Методы объекта ChartObjects

Метод	Значение
Add	Создает новую диаграмму. Синтаксис: Add (Left, Top, Width, Height); Left, Top – координаты левого верхнего угла диаграммы; Width, Height – ширина и высота диаграммы
Delete	Удаляет элемент семейства
CopyPicture	Копирует диаграмму в буфер обмена как рисунок. Синтаксис: CopyPicture (Appearance, Format, Size); Appearance – устанавливает, каким образом диаграмма копируется. Допустимые значения: xlScreen (диаграмма копируется в том виде, в котором она отображается на экране) и xlPrinter (диаграмма копируется так, как она выглядит после вывода на печать); Format – формат рисунка. Допустимые значения: xlPicture и xlBitmap
SetSourceData	Устанавливает источник данных диаграммы. Синтаксис: SetSourceData (Source, PlotBy) . Source – диапазон, на основе которого строится диаграмма; PlotBy – допустимые значения: xlColumns (данные расположены по столбцам) и xlRows (данные расположены по строкам). Пример: ActiveChart.SetSourceData Source :=Sheets (1) .Range ("A1 : F1") , PlotBy:=xlRows
Export	Экспортирует диаграмму в графический формат. Синтаксис: Export (FileName, FilterName, Interactive); FileName – имя файла, в который будет записана диаграмма в графическом формате; FilterName – имя графического фильтра в том виде, как он записан в графическом формате; Interactive – допустимые значения: True (для того чтобы показать диалоговое окно в процессе фильтрации) и False (в противном случае). Пример экспорта диаграммы в GIF-файл: ActiveChart . Export FileName := "График . gif " , FilterName := "GIF"

Окончание таблицы 12

Метод	Значение
Location	Передвигает диаграмму на новое место. Синтаксис: Location (Where, Name); Where – указывает, будет ли диаграмма располагаться на новом листе диаграмм (xlLocationAsNewSheet), внедряться как объект на рабочий лист (xlLocationAsobject) или ее местоположение будет определяться автоматически (xlLocationAutomatic); Name – имя листа диаграммы, если аргумент where принимает значение xlLocationAsNewSheet, либо имя рабочего листа, если аргумент where принимает значение xlLocationAsobject
BringToFront	Отображает диаграмму на переднем плане
SendToBack	Отображает диаграмму на заднем плане

Для построения графика на пользовательской форме применяется элемент управления Image, который используется для отображения графических файлов в формате .gif, .jpg, .ico. Свойства, используемые для управления выводом изображения на пользовательскую форму, представлены в таблице 13.

Таблица 13 – Свойства элемента управления Image

Свойство	Значение
AutoSize	True – рисунок автоматически изменяет размер для того, чтобы отобразить изображение целиком; False – в противном случае
Picture	Задаёт изображаемый графический файл. Используется совместно с функцией Loadpicture. Синтаксис: Picture = LoadPicture(полное имя файла), т. е. полное имя отображаемого графического файла
SizeMode	Устанавливает масштабирование рисунка: fmPictureSizeModeClip – не помещающиеся в границах объекта части рисунка обрезаются; fmPictureSizeModeStretch – так, чтобы он занимал поверхность объекта; fmPictureSizeModeZoom – рисунок масштабируется с сохранением относительных размеров так, чтобы он помещался целиком внутри объекта
PictureAlignment	Устанавливает расположение рисунка внутри объекта

Пример программы построения графика функции на заданном отрезке с заданным шагом на листе Excel представлен ниже:

```
Public Sub primergrafik()
Dim a As Double, b As Double, h As Double
Worksheets(1).Range("A:B").Select
Selection.Clear
Worksheets(1).ChartObject.Delete
a = Cdbl(InputBox("Введите значение a"))
b = Cdbl(InputBox("Введите значение b"))
h = Cdbl(InputBox("Введите значение h"))
j = 1
```

```

For i = a To b Step h
Worksheets(1).Range("A" & j) = i
Worksheets(1).Range("B"&j)=Sin((i*3.14)/180)/(i^2+1)
j = j + 1
Next i
график
End Sub
Public Sub график()
n=Application.CountA(Worksheets(1).Range("A:A"))
Range("A1:B" & CStr(n)).Select
Charts.Add
ActiveChart.ChartType= xlXYScatterLinesNoMarkers
ActiveChart.SetSourceData Source := Sheets ("Лист1 ") .Range("A1:B" &
CStr(n))
ActiveChart.LocationWhere:=xlLocationAsObject, Name:="Лист1"
End Sub

```

Для построения графика на пользовательской форме создается форма, на которой предусмотрены поля для ввода данных и область для вывода графика, а также кнопки для запуска построения графика, очистки формы и выхода.

Пример кода программы представлен ниже:

```

Dim g As String
Private Sub CommandButton1_Click()
Dim a As Double, b As Double, h As Double
Worksheets(1).Range("A:B").Select
Selection.Clear
'Worksheets(1).ChartObject.Delete
a = Cdbl(TextBox1.Text)
b = Cdbl(TextBox2.Text)
h = Cdbl(TextBox3.Text)
j = 1
For i = a To b Step h
Worksheets(1).Range("A" & j) = i
Worksheets(1).Range("B" & j)=Sin((i*3.14)/180)/(i^2+1)
j = j + 1
Next i
График:
Image1.Picture = LoadPicture(g)
End Sub
Public Sub График()
n = Application.CountA(Worksheets(1).Range("A:A"))
Range("A1:B" & CStr(n)).Select
Charts.Add
ActiveChart.ChartType = xlXYScatterLinesNoMarkers
ActiveChart.SetSourceData Source:=Sheets("Лист1").Range("A1:B" & CStr(n))
ActiveChart.Location Where:=xlLocationAsObject, Name:="Лист 1"

```

```

ActiveChart.Export "d:\grafik.gif"
g = "d:\grafik.gif"
End Sub
Private Sub CommandButton2_Click()
Worksheets(1).Range("A:B").Select
Selection.Clear
'Worksheets(1).ChartObject.Delete
Image1.Picture = LoadPicture("")
End Sub
Private Sub CommandButton3_Click()
UserForm1.Hide
End Sub
Private Sub UserForm_Initialize()
Image1.PictureAlignment = fmPictureAlignmentTopLeft
Image1.PictureSizeMode = fmPictureSizeModeStretch
End Sub

```

Задание 14

Построить график функции на заданном диапазоне с заданным шагом на листе Excel и на пользовательской форме. Предусмотреть обработку ошибок (таблица 14).

Таблица 14 – Варианты исходных данных

Номер варианта	Функция	Отрезок $[a, b]$	Шаг h
1	$y = 2\sin(x)\cos(x)$	$[0^\circ, 360^\circ]$	20°
2	$y = 2\sin^2(x) + \cos(x)$	$[0^\circ, 200^\circ]$	10°
3	$y = \sqrt[3]{2\sin(x)\cos(x)}$	$[-100^\circ, 100^\circ]$	20°
4	$y = \sin(2x)\cos^3(x)$	$[30^\circ, 360^\circ]$	15°
5	$y = \sqrt[3]{2\sin(x)\cos(x^2)}$	$[-45^\circ, 90^\circ]$	15°
6	$y = 2\operatorname{tg}(x + 1)$	$[0^\circ, 180^\circ]$	9°
7	$y = -\sin^3(x)\sqrt{\cos(x)}$	$[-30^\circ, 120^\circ]$	15°
8	$y = 2\operatorname{tg}(\sqrt{x})\cos(3x)$	$[0^\circ, 360^\circ]$	20°
9	$y = x\sin(x)$	$[0^\circ, 200^\circ]$	10°
10	$y = -x\cos(x)$	$[-100^\circ, 100^\circ]$	20°
11	$y = \sin^2(x) + x$	$[30^\circ, 360^\circ]$	15°
12	$y = \sqrt[3]{2\sin(x)\cos(x)}$	$[-45^\circ, 90^\circ]$	15°
13	$y = \operatorname{tg}^3(x) + \sqrt{\cos(x)}$	$[0^\circ, 180^\circ]$	9°
14	$y = \sqrt{\sin^3(x)\cos(x^2)}$	$[-30^\circ, 120^\circ]$	15°
15	$y = 2\sin(3x^2) + \operatorname{tg}(x)$	$[2, 20]$	2°

Контрольные вопросы

- 1 Назовите элемент управления для вывода диаграммы на форму.
- 2 Назовите основные свойства этого элемента управления.

15 Лабораторная работа № 23. 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.

Задание 15

Разными способами упростить следующее.

1 а) $\cos 2\alpha - \sin 4\alpha - \cos 6\alpha$;

б) $\left(\frac{3}{2x-y} - \frac{2}{2x+y} - \frac{1}{2x-5y}\right) : \frac{y^2}{4x^2-y^2}$.

2 а) $\cos^2\alpha + \cos^2\beta - 2\cos\alpha\cos\beta\cos(\alpha - \beta)$;

б) $\frac{\left(\frac{1}{a} + \frac{1}{b} - \frac{2c}{ab}\right)(a+b+2c)}{\frac{1}{a^2} + \frac{1}{b^2} - \frac{2}{ab} - \frac{4c^2}{a^2b^2}}$.

3 а) $\cos^2(\alpha + 2\beta) + \sin^2(\alpha - 2\beta) - 1$;

б) $\left(x^2 + 2x - \frac{11x-2}{3x+1}\right) : \left(x+1 - \frac{2x^2+x+2}{3x+1}\right)$.

4 а) $\frac{\sin(2\alpha-\beta)}{\cos 4\alpha} + \frac{\sin\beta}{\cos 2\alpha}$;

б) $\frac{2(x^4+4x^2-12)+x^4+11x^2+30}{x^2+6}$.

5 а) $\sin^2(\alpha - 2\beta) - \cos^2\alpha - \cos^2 2\beta$;

б) $\frac{\sqrt{(x+2)^2 - 8x}}{\sqrt{x} - 2 : \sqrt{x}}$.

6 а) $\sin^2(2\alpha - \beta) - \sin^2 2\alpha - \sin^2\beta$;

б) $\frac{\sqrt{x+1}}{x\sqrt{x+x+\sqrt{x}}} : \frac{1}{x^2-\sqrt{x}}$.

7 а) $\cos 2\alpha - \sin 4\alpha - \cos 6\alpha$;

б) $\sqrt[4]{6x(5+2\sqrt{6})} \cdot \sqrt{3\sqrt{2x} - 2\sqrt{3x}}$.

8 а) $1 - \sin\left(\frac{\alpha}{2} - 3\pi\right) - \cos^2\frac{\alpha}{4} + \sin^2\frac{\alpha}{4}$;

б) $\frac{a^3 - 2a^2 + 5a + 26}{a^3 - 5a^2 + 17a - 13}$.

Контрольные вопросы

1 Что такое переменная?

2 Что такое ранжированная переменная?

3 Что такое функция?

16 Лабораторная работа № 24. 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.

Задание 16

1 Ознакомиться с основными теоретическими положениями.

2 Выполнить задание, полученное у преподавателя.

3 Оформить отчет в виде набора файлов.

Вычислить значение кусочно-непрерывной функции для значений аргумента $x = a$, $x = b$ и $x = -c$ с использованием программного фрагмента.

$$y(x) = \begin{cases} e \cdot x^2, & \text{если } x > g; \\ d \cdot x^3, & \text{если } x < 0; \\ f \cdot x & \text{в остальном случае.} \end{cases}$$

Таблица 15 – Варианты исходных данных

Номер варианта	a	b	c	d	e	f	g
1	2	6	2	4	1	3	4
2	3	7	3	4	2	3	5
3	4	8	4	4	3	4	6
4	5	9	5	5	4	4	7
5	6	10	6	5	5	5	8
6	7	11	7	5	6	5	9
7	8	12	8	6	7	6	10
8	9	13	9	4	8	1	12
9	10	14	10	5	9	2	13
10	11	15	11	6	1	3	14
11	12	16	12	7	2	4	15
12	12	17	13	8	3	5	16

С использованием программного фрагмента вычислить сумму

$$1/a^2 + 1/b^2 + \dots + 1/c^2.$$

Таблица 16 – Варианты исходных данных

Номер варианта	a	b	c
1	1	2	100
2	1	3	100
3	1	4	100
4	1	5	100
5	1	6	100
6	1	7	100

Окончание таблицы 16

Номер варианта	a	b	c
7	1	8	100
8	1	9	100
9	1	10	100
10	1	11	100
11	1	12	100
12	1	13	100

Контрольные вопросы

- 1 Как создать программный фрагмент в Mathcad?
- 2 Какие команды содержит панель программирования?
- 3 Принцип работы команды ветвления.
- 4 Общий вид оператора *if* и порядок его работы.
- 5 Принцип работы команд цикла.
- 6 Общий вид оператора *for* и порядок его работы.
- 7 Общий вид оператора *while* и порядок его работы.
- 8 В чем отличие команд цикла *for* и *while*?
- 9 Какие виды графиков существуют в Mathcad?
- 10 Как форматировать график?
- 11 Как построить несколько независимых кривых на одном поле графика?

17 Лабораторная работа № 25. Использование пакета Mathcad для решения математических задач

Цель работы: получение навыков нахождения решений системы линейных и нелинейных уравнений в программе 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

Все методы решения систем линейных алгебраических уравнений можно разделить на две основных группы: прямые (метод Крамера, метод Гауса и т. п.) и итеративные методы. При использовании прямых методов расчеты можно вести как численно, так и символично. Итеративные методы применяются в численных решениях.

Для решения систем линейных и нелинейных уравнений используется «блок решений», который начинается из ключевого слова Given и заканчивается вызовом функции Find. Между ними находятся уравнение. Всем неизвестным в

уравнении должны быть присвоены начальные значения. В уравнении, для которого необходимо найти решение, нужно использовать знак логического равенства $=$ с панели инструментов Evaluation. В качестве аргументов функции должны быть неизвестные, которые необходимо найти.

Решение системы линейных уравнений с помощью встроенной функции $Lsolve(A,b)$ возвращает вектор решений b . Матрица A – квадратная невырожденная, вектор b – вектор правых частей в системе уравнений.

С помощью символьного процессора Mathcad можно получать аналитические решения системы уравнений, используя оператор Solve. В этом случае система должна быть занесена в виде вектора в левый маркер оператора. Переменные, значения которых отыскиваются, следует вводить через запятую в правый маркер оператора Solve. Ответ будет возвращен в виде матрицы, в строках которой будут записаны найденные значения неизвестных системы уравнений.

Приближенные решения системы уравнений можно получить с использованием встроенной функции $Minerr(x_1, \dots)$. Эта функция подобная по своей работе к функции Find, однако она имеет другие условия для завершения итеративного процесса поиска решений. Функция Minerr позволяет находить решение в том случае, когда их не находит функция Find.

Задание 17

Преобразовать с помощью команды *Expand* следующие выражения (таблица 17).

Таблица 17 – Варианты исходных данных

Номер варианта	Функция	Номер варианта	Функция
1	$(\sin x - 2)^2 (x + y)^2$	5	$\cos 3x (a + b)^2$
2	$\operatorname{tg} 2x (a + b + c)^2$	6	$\sin^4 x (a - b)^4$
3	$\cos 2x (a + b - c)^2$	7	$\cos^3 x (a + b)^3$
4	$\sin 3x (a - b - c)^2$	8	$\operatorname{tg} 3x (a + b)^5$

Преобразовать с помощью команды *Factor* следующие выражения (таблица 18).

Таблица 18 – Варианты исходных данных

Номер варианта	Функция	Номер варианта	Функция
1	$1 - 2\sin x + \sin^2 x$	5	$a^2 - b^2$
2	$x^3 - y^3$	6	$a^4 - 4a^3b + 6a^2b^2 - 4ab^3 + b^4$
3	$a^4 + a^2b^2 + b^4$	7	$a^2 - 2a\sin x + \sin^2 x$
4	12456	8	$1 - x^4$

Контрольные вопросы

- 1 Назовите методы решения систем линейных алгебраических уравнений.
- 2 Назовите способы решения систем линейных и нелинейных уравнений.

3 Назовите встроенные функции для решения систем линейных и нелинейных уравнений.

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.

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

3 **Коваленко, В. В.** Проектирование информационных систем: учебное пособие / В. В. Коваленко. – 2-е изд., перераб. и доп. – Москва: ИНФРА-М, 2021. – 357 с.