

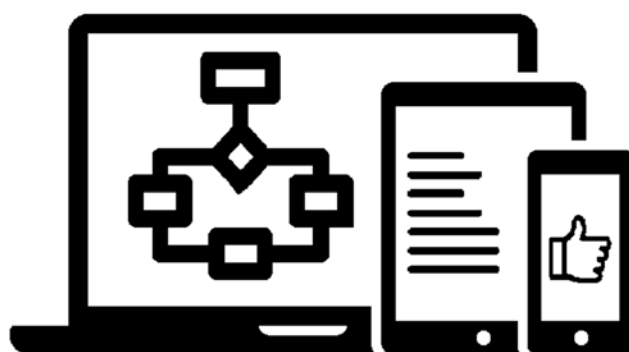
МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ИНФОРМАТИКА

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
15.03.01 «Машиностроение»
очной формы обучения*

Часть 2



Могилев 2022

УДК 004
ББК 32.97
И74

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «27» октября 2022 г., протокол № 4

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

Рецензент канд. техн. наук, доц. В. М. Ковальчук

В методических рекомендациях кратко изложены теоретические сведения, необходимые для выполнения лабораторных работ. Рекомендации составлены в соответствии с учебной программой по дисциплине «Информатика» для студентов направления подготовки 15.03.01 «Машиностроение» очной формы обучения.

Учебно-методическое издание

ИНФОРМАТИКА

Часть 2

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

Подписано в печать 27.12.2022 . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. 1,16. Уч.-изд. л. 1,06 . Тираж 21 экз. Заказ № 1252.

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, 2022

Содержание

Введение	4
1 Программирование линейных алгоритмов.....	6
2 Условные операторы.....	11
3 Циклы	12
4 Одномерные массивы	15
5 Сортировка данных	16
6 Двумерные массивы.....	16
Список литературы	17

Введение

Цель изучения дисциплины «Информатика» заключается в приобретении студентами теоретических основ и практических навыков работы пользователя в среде MS Office, в том числе и навыков программирования на языке VBA, а цель настоящей разработки – в создании методических рекомендаций, позволяющих закрепить теоретические знания и приобрести практические навыки работы и программирования в среде VBA.

Чаще всего основным офисным пакетом, в котором пользователь создаёт свои приложения, является MS Excel. Поэтому в настоящих методических рекомендациях рассматривается программирование на языке Visual Basic for Application (VBA) в Excel.

Язык VBA предназначен для автоматизации разработки документов при решении задач, использующих приложения Excel, Word, Access, PowerPoint, Outlook, – начиная с проектирования простых документов и заканчивая автоматизацией документооборота. Язык VBA является языком программирования высокого уровня, в нем широко применяются элементы объектно-ориентированного и событийно-управляемого программирования. При разработке приложений в VBA можно использовать средства процедурного и объектно-ориентированного программирования [1].

Процедурное программирование является отражением архитектуры традиционных ЭВМ, которая была предложена фон Нейманом в 1940-х гг. Теоретической моделью процедурного программирования служит алгоритмическая система под названием «машина Тьюринга».

Программа на процедурном языке программирования состоит из последовательности операторов (инструкций), задающих процедуру решения задачи. Основным является оператор присваивания, служащий для изменения содержимого областей памяти. Выполнение программы сводится к последовательному исполнению ее операторов с целью преобразования значений исходных данных в результаты. Примерами процедурных языков программирования являются Assembler, Pascal, C, PL/1, FORTRAN.

Объектно-ориентированное программирование (ООП) – это совокупность подходов, методов, стратегий, идей и понятий, определяющая стиль написания программ, в котором основными концепциями являются понятия объектов и классов.

Объект – это некоторая сущность, обладающая определённым состоянием и поведением, имеет заданные значения атрибутов и операций. Как правило, при рассмотрении объектов выделяется то, что они принадлежат одному или нескольким классам, которые в свою очередь определяют поведение объекта.

Класс – это объединение данных и обрабатывающих их процедур и функций. Данные называются также переменными класса, а процедуры и функции – методами класса.

Переменные определяют свойства объекта, а совокупность их значений – его состояние.

Каждый класс имеет определенный набор событий, которые могут возникать при работе с объектами класса, чаще всего при определенных действиях пользователя, иногда как результат действия системы.

Наследование позволяет описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом.

Инкапсуляция – это свойство языка программирования, позволяющее объединить данные и код в объект и скрыть реализацию объекта от пользователя. При этом пользователю предоставляется только интерфейс объекта, т. е. способ взаимодействия с объектом. Пользователь может взаимодействовать с объектом только через этот интерфейс.

Полиморфизм – это явление, при котором один и тот же программный код выполняется по-разному в зависимости от того, объект какого класса используется при вызове данного кода.

Абстракция данных – подход к обработке данных по принципу чёрного ящика. Позволяет работать с объектами, не вдаваясь в особенности их реализации.

Методические рекомендации предназначены для изучения объектно-ориентированного программирования на языке VBA в среде табличного процессора MS Excel, который включает основные средства создания приложений, структуры данных и управляющие структуры.

При описании операторов в данных методических рекомендациях используется общий синтаксис, в котором в угловых скобках «< >» указывается параметр, задаваемый пользователем, а в квадратных «[]» – необязательный параметр, который может быть использован или опущен.

Дополнительные сведения по программированию содержатся в литературных источниках, приведенных в разделе «Список литературы».

При выполнении лабораторных работ студент должен использовать среду программирования VBA текстового процессора Excel, которая поддерживает технологию объектно-ориентированного программирования.

1 Программирование линейных алгоритмов

Цель работы: изучить основы языка программирования VBA.

1.1 Краткие теоретические сведения

Типы данных определяют, каким образом величины хранятся в памяти компьютера. Наиболее распространенные типы данных приведены в таблице 1.1.

Таблица 1.1 – Основные типы данных

Тип данных	Диапазон
Byte (целый без знака)	От 0 до 255
Boolean (логический)	True или False
Integer (целый)	От -32768 до 32767
Long (длинный целый)	От -2147483648 до 2147483647
Single (с плавающей точкой одинарной точности)	От -3.402823E+38 до -1.401298E-48. От 1.401298E-45 до 3.402823E+38
Double (с плавающей точкой двойной точности)	От -1,79769313486232E308 до -4,94065645841247E-324. От 4,94065645841247E-324 до 1,79769313486232E308
Date (дата и время)	От 01.01.100 до 31.12.9999
String (строка)	От 0 до 65535 символов
Variant	Любой тип, за исключением типа Object

1.1.1 Переменные. Переменная – это поименованная область в памяти компьютера, которая может изменять свое значение в ходе выполнения алгоритма. Она всегда обозначается символическим именем (X, A, R5 и т. п.), начинающимся с буквы и не содержащим пробелов. Тип переменной соответствует типу представляемых ею данных.

Переменные предназначены для хранения данных в оперативной памяти. В программе переменные, перед использованием их, желательно объявлять с помощью оператора Dim (Dim <ИмяПеременной>[As <тип>]).

В языке VBA имеется возможность работать с динамическими массивами, массивами переменной длины, которые объявляются без указания их размера:

Dim <переменная> () [As <тип>].

Затем в подходящем месте программы записывается оператор

Redim [Preserve] <переменная> (<макс. размер>) [<As тип>],

где «макс. размер» строится по тем же правилам, что и границы индексов при объявлении массива. Ключевое слово *Preserve* указывает, что при переопределении массива предыдущие значения не уничтожаются. *Redim* можно использовать неограниченное количество раз, каждый раз меняя размерность массива.

Неявное объявление. В этом способе определения переменных никакие инструкции для объявления вообще не используются. Когда транслятор VBA

первый раз встречает в коде программы необъявленную переменную, он назначает ей тип `Variant`, и переменная может принимать значение любого типа.

1.1.2 Константы. Константы не изменяют своего значения в ходе выполнения программы. Они могут обозначаться собственным значением (числа 10, 3.5 и т. д.) или символическими именами. Для объявления констант используется оператор Const (Const <ИмяКонстанты> [As <Тип>] = <Выражение>).

1.1.3 Массивы. Представляет собой структуру, все элементы которой имеют одинаковый тип. Они могут быть одномерными и многомерными. Количество размерностей массива может достигать 64. В VBA массив объявляется оператором Dim (Dim ИмяМассива([Индексы]) [As ТипДанных]).

1.2 Выражения

Выражение – это запись, определяющая последовательность действий над величинами. Выражения строятся из операндов (константы, переменные, элементы массивов и функции), соединенных знаками операций.

В языке VBA используются следующие виды выражений: арифметические; выражения отношения; логические и строковые.

1.2.1 Используемые операции.

В арифметических выражениях:

сложение (+); вычитание (-);
 умножение (*); деление (/);
 возведение в степень (^); целочисленное деление (\);
 A mod B – остаток от деления по модулю.

В логических выражениях:

NOT – операция "не" (инверсия);
 OR – операция "или" (логическое сложение);
 AND – операция "и" (логическое умножение);
 XOR – операция "исключающее или";
 EQV – операция "проверка логического равенства двух выражений";
 IMP – операция "импликация".

В выражениях отношения (таблица 1.2).

Таблица 1.2 – Операции отношения

Опера- ция	Комментарий	Опера- ция	Комментарий	Опера- ция	Комментарий
>	Больше	<	Меньше	=	Равно
>=	Больше либо равно	<=	Меньше либо равно	<>	Не равно

Порядок выполнения операций в выражениях определяется старшинством операций, которое приведено в таблице 1.3.

Таблица 1.3 – Старшинство операций (в порядке убывания приоритета)

Приоритет	Операция	Приоритет	Операция	Приоритет	Операция
1	Операции в скобках	5	*, /, \, mod	9	And
2	Вычисление функции	6	+, −	10	Or
3	^	7	=, >, <, >=, <=, <>	11	Xor
4	Смена знака	8	Not		

1.2.2 Математические функции. В выражениях могут использоваться стандартные математические функции, основные из которых приведены в таблице 1.4.

Таблица 1.4 – Стандартные математические функции VBA

Обращение	Функция
Abs(x)	Модуль аргумента
Atn(x)	Арктангенс (радианы)
Cos (x)	Косинус
Exp(x)	e ^x — экспонента
Int(x)	Целая часть x, полученная отбрасыванием дробной части
Fix(x)	Число, округленное до ближайшего меньшего целого
Log(x)	Натуральный логарифм
Sin(x)	Синус
Sqr(x)	Корень квадратный
Tan(x)	Тангенс числа

1.2.3 Функции преобразования типов. Выполняют преобразование переменной некоторого типа в заданный тип. Чаще всего выполняется преобразование из строки символов в число и числа в строку символов (таблица 1.5).

Таблица 1.5 – Функции преобразования типов

Функция	Назначение
Val(строка)	Возвращает числа, содержащиеся в строке, как числовое значение соответствующего типа
Str(число)	Возвращает значение типа variant (String), являющееся строковым представлением числа

В качестве допустимого разделителя функция Str воспринимает только точку. При наличии другого десятичного разделителя следует использовать функцию CStr. Остальные функции преобразования из данного типа в указанный приведены в таблице 1.6.

Таблица 1.6 – Функции преобразования типов

Функция	Тип	Функция	Тип	Функция	Тип
CBool	Boolean	CDbl	Double	CSng	Single
CByte	Byte	CInt	Integer	CStr	String
CDate	Date	CLng	Long		

Функции проверки типа позволяют определить, является ли переменная выражением определенного типа (таблица 1.7).

Таблица 1.7 – Функции проверки типа

Функция	Проверка
IsNumeric(x)	Является ли переменная числовым значением
IsNull(x)	Является ли переменная пустым значением
IsError(x)	Является ли переменная кодом ошибки

1.2.4 Логические выражения. Логические выражения в результате вычисления принимают логические значения True (Истина) или False (Ложь). Операндами логического выражения могут быть логические константы, переменные логического типа, отношения. В VBA чаще используют четыре логические операции: отрицание – NOT, логическое умножение – AND, логическое сложение – OR, исключающее «или» – XOR. Результаты логических операций для различных значений операндов приведены в таблице 1.8.

Таблица 1.8 – Результаты логических операций

A	B	not A	A and B	A or B	A xor B
True	True	False	True	True	False
True	False	False	False	True	True
False	False	True	False	False	False
False	True	True	False	True	True

1.3 Операторы (инструкции) VBA

1.3.1 Оператор присваивания. Это инструкция VBA, выполняющая математическое вычисление и присваивающая результат переменной или объекту. Формат оператора присваивания:

<имя переменной или свойство> = <выражение>

1.3.2 Операторы (функции) ввода-вывода. В языке VBA для ввода значений переменных и массивов можно использовать функцию *InputBox*, а для вывода – функцию *MsgBox*.

Функция *InputBox* показывает диалоговое окно ввода, ожидает ввода текста и возвращает содержимое введенного текста после закрытия окна.

Синтаксис функции *InputBox*

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

где *prompt* – строковое выражение длиной до 1024 символов, отображаемое как пояснительный текст в диалоговом окне. Чтобы этот текст разместить более чем в одной строке, в месте разрыва строки следует вставить символы возврат каретки *Chr(13)* и перевод строки *Chr(10)*; *title* – текст заголовка диалогового окна. Если он отсутствует, то в заголовок окна выводится имя приложения, например *Microsoft Excel*; *default* – текст по умолчанию, возвращаемый в программу как значение по умолчанию. Если он отсутствует, то поле ввода диалогового окна ввода будет пустым.

Функция *MsgBox* показывает сообщение *prompt* в диалоговом окне, ожидает выбора пользователем одной из кнопок и возвращает значение, указывающее, какая кнопка была выбрана. Синтаксис функции *MsgBox*

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

где *prompt* – аналогично функции *InputBox*; *buttons* – выражение, определяющее число и тип кнопок выбора в диалоговом окне и кнопку по умолчанию.

Отладочная печать. Используется для вывода результатов в окно *Immediate* с помощью оператора *Debug.Print*. Это окно можно отобразить, прибегнув к комбинации клавиш *Ctrl + G* или команды *View / Immediate Window*.

1.3.3 Ввод данных с листа Excel и вывод на лист Excel. Для ввода данных с листа Excel 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, которые должны быть оформлены в стиле A1, и возвращает объект *Range*. Ссылка на единичную ячейку, использованная в качестве аргумента, возвращает объект *Range* для единичной ячейки. Две ссылки на единичные ячейки возвращают объект *Range* для прямоугольной области, заключенной между этими двумя ячейками (таблица 1.9).

Таблица 1.9 – Примеры использования метода *Range*

Оператор	Комментарий
<code>X = Worksheets("Лист1").Range("B1").Value</code>	Переменной X присваивается значение ячейки B1 из листа 1
<code>Worksheets("Лист1").Range("B1").Value = X</code>	В ячейку B1 листа 1 выводится значение переменной X
<code>Worksheets("Лист1").Range("C1","D6").Value = 27</code>	В ячейки C1 и D6 листа 1 выводится число 27
<code>Worksheets("Лист1").Range("B7:C9").Value = 54</code>	В диапазон ячеек "B7:C9" листа 1 выводится число 54

Метод *Cells*, получая в качестве аргументов два целых числа, возвращает объект, содержащий единичную ячейку. Аргументы определяют номера строки и столбца выбранной ячейки (таблица 1.10).

Таблица 1.10 – Примеры использования метода *Cells*

Оператор	Комментарий
<code>A=Worksheets(1).Cells(1,2).Value</code>	Переменной A присваивается значение из ячейки B1 (строка 1, столбец 2) листа 1
<code>Worksheets(1).Cells(4,2).Value= X</code>	В ячейку B4 (строка 4, столбец 2) записывается значение переменной X

1.4 Практическое задание

По заданию преподавателя реализовать простейшую программу.

Контрольные вопросы

- 1 Как объявить переменную, которая может иметь целые значения в диапазоне от 0 до 100?
- 2 Как в программе присвоить значение переменной?
- 3 Как показать результат вычислений?

2 Условные операторы

Цель работы: научиться работать с условными операторами.

2.1 Краткие теоретические сведения

2.1.1 Условный оператор *If*. Изменяет естественную последовательность выполнения операторов программы, т. е. в зависимости от истинности логического выражения (условия) он выполняет одну или другую последовательность операторов. Синтаксис оператора *If*

```

If <условие> Then <группа операторов>
или
If <условие> Then
    <группа операторов 1>
Else
    <группа операторов 2>
End If
или
If <условие 1> Then
    <группа операторов 1>
ElseIf <условие 2> Then
    <группа операторов 2>
Else
    <группа операторов 3>
End If

```

2.1.2 Оператор выбора *Select Case*. Выполняет один или несколько операторов в зависимости от значения выражения. Синтаксис оператора

```

Select Case <выражение>
    [Case <список 1>
        <Блок операторов 1>]
    [Case <список 2>
        <Блок операторов 2>]
    ...
    [Case <список n2>
        <Блок операторов n>]
    [Case Else
        <Блок операторов n>]

```

End Select

Здесь <выражение> – любое числовое или строковое выражение;
 <список 1>, ..., <список n> – список значений в форме

<выражение>; либо

<выражение> *To* <выражение>; либо

Is <операция сравнения> <выражение>

Если значение <выражение> совпадает с одним из значений <список i>, то выполняется <Блок операторов i> и управление передается оператору, следующему за строкой *End Select*.

2.2 Практическое задание

По заданию преподавателя реализовать простейшую программу с условным оператором.

Контрольные вопросы

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

2 В каких случаях в программе используется полный условный оператор? Как он оформляется? Как он работает (что происходит при его выполнении)?

3 В каких случаях в программе используется неполный условный оператор? Как он оформляется?

3 Циклы

Цель работы: изучить основные приемы работы с циклами/

3.1 Общие сведения

Операторы цикла. Операторы цикла позволяют повторить блок операторов программы до тех пор, пока верно некоторое условие, или заданное число раз. В языке VBA используются следующие операторы цикла: *For ... Next*; *Do ... Loop*; *While ... Wend*; *For Each ... Next*/

Оператор *For ... next*. Синтаксис оператора

For i = in To ik [Step_h]

[блок операторов]

[Exit For]

[блок операторов]

Next [i]

Здесь i – управляющая переменная цикла; in, ik – начальное и конечное значения управляющей переменной цикла; h – шаг ее изменения.

Оператор *For ... next* выполняет циклически (повторяет) группу операторов, заключенных между строкой *For i = in To ik* и строкой *Next i*. При этом количество повторений будет равно $(ik - in) / Step_h + 1$.

Возможен досрочный выход из цикла с помощью оператора *Exit For*, который может использоваться, например, совместно с оператором *If*.

Оператор *For Each ... next*. Синтаксис оператора

```
For Each a In group
[блок операторов]
[Exit For]
[блок операторов]
Next [element]
```

Здесь *a* – переменная, используемая для обозначения элементов коллекции или группы *group*; *group* – имя коллекции объектов, группы элементов или массива. Данный оператор на каждом шаге своего выполнения присваивает переменной *a* очередной элемент из коллекции *group* и выполняет [блок операторов]. Здесь также возможен досрочный выход из цикла с помощью оператора *Exit For*.

Оператор *Do ... Loop* повторяет блок операторов до тех пор, пока условие верно, или до тех пор, пока условие не станет верным. Данный оператор может использоваться в двух формах. Первая форма называется формой с предусловием, т.к. в ней условие выхода из цикла указывается в начале, после *Do [{While | Until}]*:

```
Do [{While | Until} <условие>]
[Блок операторов]
[Exit Do]
[Блок операторов]
Loop
```

Вторая форма называется формой с постусловием, т.к. в ней условие выхода из цикла указывается в конце, после *Loop [{While | Until}]*:

```
Do
[Блок операторов]
[Exit Do]
[Блок операторов]
Loop [{While | Until} <условие>]
```

Параметр *While* означает: «пока условие верно» область цикла будет выполняться. Параметр *Until* означает: «пока условие не станет верным» область цикла будет выполняться.

Оператор *While ... Wend* выполняет в цикле последовательность операторов до тех пор, пока верно условие. Синтаксис оператора

```
While <условие>
[Блок операторов]
Wend,
```

где <условие> – логическое выражение или выражение отношения.

Таблица 3.1 – Варианты заданий для выполнения лабораторной работы

Номер варианта	Задание	Номер варианта	Задание
1	$y = x^3 + a \cdot \sin^2(x^3 + 0,15) - e^{2 \cdot x};$ $0,2 \leq x \leq 2,8; \quad hx=0,2$	9	$y = a \cdot x^2 + b \cos^3(x^2 - 0,1) - e^x;$ $0,1 \leq x \leq 2,5; \quad hx=0,2$
2	$y = a \cdot x^2 + b \cdot x \cdot \operatorname{tg}(x^2 - 0,15) - e^{x/2};$ $0,2 \leq x \leq 2,4; \quad hx=0,1$	10	$y = b \cdot x^3 + a \cdot x^3 \cdot \sin(x^3 + 0,1) - e^x;$ $0,1 \leq x \leq 2,9; \quad hx=0,1$
3	$y = \frac{a \cdot x^2}{2} + b \cdot x \sin^2(x^{1,2} + 0,1) + e^x;$ $0,2 \leq x \leq 2,6; \quad hx=0,2$	11	$y = b + a \cdot x^3 \cdot \cos^2(x^{1,3} + 0,1) - e^x;$ $0,1 \leq x \leq 1,5; \quad hx=0,1$
4	$y = \frac{a-1}{b} + x \cdot \cos^{2,5}(x^2 + 0,15) + e^x;$ $0,2 \leq x \leq 4,0; \quad hx=0,12$	12	$y = a \cdot x^{2,5} + b \cdot \sin^2(x^{2,1} + 0,1) - e^x;$ $0,1 \leq x \leq 2,55; \quad hx=0,1$
5	$y = \frac{-x^2}{a \cdot b} + x \cdot \operatorname{tg}^2(x^3 + 0,15) + e^{2 \cdot x};$ $0,2 \leq x \leq 2,8; \quad hx=0,2$	13	$y = \frac{a}{x^2} + \frac{x}{b} \cdot \operatorname{tg}^2(x^{1,3} + 0,1) - e^{2 \cdot x};$ $0,1 \leq x \leq 3,1; \quad hx=0,1$
6	$y = \frac{a \cdot b}{x^2} + x \cdot \sin^2(x^3 + 0,2) + e^{2 \cdot x};$ $0,2 \leq x \leq 3,2; \quad hx=0,2$	14	$y = \frac{-a}{x^2} \cdot \operatorname{tg}^2 x - x - e^{2 \cdot x};$ $0,1 \leq x \leq 2,9; \quad hx=0,1$
7	$y = x^3 - b \cdot x^2 \cdot \operatorname{tg}^2(x^{1,3} + 0,1) - e^x;$ $0,2 \leq x \leq 3,4; \quad hx=0,2$	15	$y = \frac{a \cdot b}{x^2} + \operatorname{tg}^3(x^{2,3} + 0,1) - e^{2 \cdot x};$ $0,1 \leq x \leq 3,1; \quad hx=0,1$
8	$y = x^3 + a \cdot x \cdot \cos^2(x^{1,2} + 0,3) - e^x,$ $0,2 \leq x \leq 3,8; \quad hx=0,2$	16	$y = \frac{a}{b \cdot x^2} + \sin^3(x + 0,2)^2 - e^{3x},$ $0,1 \leq x \leq 3,4; \quad hx=0,1$

3.2 Практическое задание

Постановка задачи. Составить процедуру вычисления значений y для каждого значения переменной x , изменяющейся в заданных пределах и с заданным шагом изменения. Варианты заданий для выполнения работы приведены в таблице 3.1.

Контрольные вопросы

- 1 Прокомментируйте назначение оператора DO.
- 2 Как работает оператор цикла DO с предусловием?
- 3 Как работает оператор цикла с постусловием?

- 4 Для чего используется оператор отладочной печати?
 5 В какой объект выводятся данные отладочной печати?
 6 Сформулируйте условия выхода из цикла DO.

4 Одномерные массивы

Цель работы: научиться работать с одномерными массивами.

Задание. Составить процедуру Sub() формирования массива X(25) вещественного типа и вычисления параметра, указанного в таблице 4.1.

Таблица 4.1 – Варианты заданий для выполнения лабораторной работы № 4

Номер варианта	Задание
1	Сформировать элементы массива по правилу $x_i = \cos(r)$. Определить максимальный элемент массива и его порядковый номер
2	Сформировать элементы массива по правилу $x_i = \sin(r)$. Вычислить сумму и среднее арифметическое положительных элементов массива
3	Сформировать элементы массива по правилу $x_i = (i+1)\cos(r)$. Найти минимальное, максимальное и среднее арифметическое значения элементов массива
4	Сформировать элементы массива по правилу $x_i = \sin(r) - \cos(r)$. Создать двумерный массив у, первый столбец которого содержит положительные элементы массива х, а второй – отрицательные
5	Сформировать элементы массива по правилу $x_i = \sin(2r) - \cos(r)$. Переписать в отдельный массив элементы с номерами, кратными 3, и определить их количество
6	Сформировать элементы массива по правилу $x_i = \sin(1,5r) + \cos(r)$. Переписать в массив Y элементы с четными номерами и определить их количество
7	Сформировать элементы массива по правилу $x_i = \sin(r) - \cos(r)$. Поменять местами максимальный и минимальный элементы массива
8	Сформировать элементы массива по правилу $x_i = \sin(r) - \cos(r)$. Найти сумму абсолютных значений максимального и минимального элементов
9	Сформировать элементы массива по правилу $x_i = \sin(r) - \cos(r)$. Заменить все отрицательные элементы массива на 0
10	Сформировать элементы массива по правилу $x_i = \sin(3r) - \cos(r)$. Определить минимальный элемент массива и его порядковый номер
11	Сформировать элементы массива по правилу $x_i = \sin(3r) + \cos(2r)$. Определить сумму отрицательных элементов массива и их количество
12	Сформировать элементы массива по правилу $x_i = 2\sin(r) + \cos(2r)$. Найти отклонение каждого элемента массива от их среднего арифметического значения
13	Сформировать массив по правилу $x_i = \sin(2r) + 2\cos(r)$. Определить сумму элементов массива с четными номерами, переписать их в отдельный массив у

Окончание таблицы 4.1

14	Сформировать элементы массива по правилу $x_i = 3\sin(r) - \cos(r)$. Переписать в массив Y элементы массива X, меньшие их среднего арифметического значения
15	Сформировать элементы массива по правилу $x_i = \sin(2r) + 2\cos(r)$. Посчитать сумму элементов, стоящих в массиве на нечетных местах, и определить их среднее арифметическое значение
16	Сформировать элементы массива по правилу $x_i = 2\sin(r) - \cos(r)$. Найти сумму квадратов отклонений каждого элемента массива от их среднего арифметического (среднее квадратическое отклонение) значения
<i>Примечание</i> – Величина r – это формируемое функцией Rnd псевдослучайное число в интервале 0...1	

Контрольные вопросы

- 1 Что такое массив?
- 2 Что такое размер и размерность массива?
- 3 Как обратиться в процедуре или функции к элементу массива?
- 4 Прокомментируйте алгоритм сортировки массива.
- 5 Прокомментируйте алгоритм перемножения матриц.

5 Сортировка данных

Цель работы: научиться выполнять сортировку данных.

Задание. Составить процедуру сортировки одномерного массива X(25) в порядке возрастания его элементов. Инициализацию массива выполнить с использованием функции Rnd.

6 Двумерные массивы

Цель работы: научиться работать с двумерными массивами.

Задание. Составить процедуру, выполняющую перемножения двух матриц $|C| = |A| * |B|$. Размерности матриц: A(n, m), B(m, k). Размерность матрицы C определить самостоятельно.

Список литературы

- 1 **Гвоздева, В. А.** Информатика, автоматизированные технологии и системы: учебник / В. А. Гвоздева. – Москва: ФОРУМ; ИНФРА-М, 2021. – 542 с.
- 2 Информатика. Базовый курс: учебное пособие / Под ред. С. В. Симоновича. – 3-е изд. – Санкт-Петербург: Питер, 2017. – 640 с.: ил.
- 3 **Гуриков, С. Р.** Введение в программирование на языке Visual Basic for Applications (VBA): учебное пособие / С. Р. Гуриков. – Москва: ИНФРА-М, 2021. – 317 с.
- 4 **Скитер, Н. Н.** Информационные технологии: учебное пособие / Н. Н. Скитер, А. В. Костикова. – Волгоград: ВолгГТУ, 2019. – 96 с.
- 5 **Комолова, Н. В.** Программирование на VBA в Excel 2016: самоучитель / Н. В. Комолова, Е. С. Яковлева. Санкт-Петербург: БХВ-Петербург, 2017. - 432 с.
- 6 **Власовец, А. М.** Введение в VBA: учебное пособие / А. М. Власовец. – Санкт-Петербург: СПбГУЭФ, 2010. – 135 с.
- 7 **Биллинг, В. А.** VBA и Office 2000. Офисное программирование / В. А. Биллинг. – Москва: Русская редакция, 1999. – 480 с.
- 8 **Гарнаев, А. Ю.** Использование MS Excel и VBA в экономике и финансах / А. Ю. Гарнаев. – Санкт-Петербург: БХВ- Петербург, 2002. – 336 с.
- 9 **Гарнаев, А. Ю.** Самоучитель VBA / А. Ю. Гарнаев. – Санкт-Петербург: БХВ-Петербург, 2002. – 512 с.
- 10 **Васильев, А.** VBA в Office 2000: учебный курс / А. Васильев. – Санкт-Петербург: Питер, 2002. – 412 с.
- 11 **Кузьменко, В. Г.** VBA 2002 / В. Г. Кузьменко. – Москва: БИНОМ, 2002. – 432 с.