

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

# ИНФОРМАТИКА

*Методические рекомендации к лабораторным работам  
для студентов направления подготовки 27.03.05 «Инноватика»  
и 41.03.01 «Зарубежное регионоведение» дневной формы обучения*

## Часть 2



УДК 004.42  
ББК 32.97  
И74

Рекомендовано к изданию  
учебно-методическим отделом  
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «24» сентября 2024 г., протокол № 2

Составитель канд. техн. наук, доц. Э. И. Ясюкович

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

Методические рекомендации содержат базовые сведения по основам информатики, некоторые приемы работы по офисным информационным технологиям, а также задания для выполнения лабораторных работ.

Учебное издание

ИНФОРМАТИКА

Часть 2

Ответственный за выпуск	В. В. Кутузов
Корректор	А. А. Подошевка
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 16 экз. Заказ №

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

© Белорусско-Российский  
университет, 2024

## Содержание

Введение.....	4
1 Основы языка программирования VBA .....	6
1.1 Типы данных.....	6
1.2 Выражения. ....	7
1.3 Операторы (инструкции) VBA.....	9
1.4 Процедуры и функции .....	13
1.5 Обработка строк .....	14
1.6 Интерфейс редактора VBA .....	15
2 Методические указания к выполнению лабораторных работ .....	18
2.1 Лабораторная работа № 1. Программирование с использованием объектов Excel. Линейная программа .....	19
2.2 Лабораторная работа № 2. Разветвляющаяся программа .....	20
2.3 Лабораторная работа № 3. Оператор выбора Select Case .....	22
2.4 Лабораторная работа № 4. Массивы и циклы For...Next .....	23
2.5 Лабораторная работа № 5. Циклы While...Wend и Do...Loop.....	25
2.6 Лабораторная работа № 6. Численные методы.....	26
2.7 Лабораторная работа № 7. Пользовательские формы. Элементы управления VBA: Label, TextBox, ComboBox, CommandButton .....	27
2.8 Лабораторная работа № 8. Элемент управления Переключатель.....	28
2.9 Лабораторная работа № 9. Обработка строк.....	29
Список литературы.....	31

## 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 указаний, 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 возни-

кать 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 методических рекомендациях используется общий синтаксис, в котором в угловых скобках «< >» указывается параметр, задаваемый пользователем, а в квадратных «[ ]» – указывается 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 будет использовать среду программирования VBA текстового процессора Excel, которая поддерживает технологию объектно-ориентированного программирования.

# 1 Основы языка программирования VBA

## 1.1 Типы данных

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:

$$\text{Dim } \langle \text{переменная} \rangle () [\text{As } \langle \text{тип} \rangle]$$

Затем в подходящем месте программы записывается оператор

$$\text{ReDim } [\text{Preserve}] \langle \text{переменная} \rangle (\langle \text{макс. размер} \rangle) [\langle \text{As } \text{тип} \rangle],$$

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 встречается 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.

Операции отношения представлены таблице 1.2.

Таблица 1.2 – Операции отношения

Операция	Комментарий	Операция	Комментарий	Операция	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.

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

*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 1.3 – Старшинство операций (в порядке убывания приоритета)

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

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

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.

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



Таблица 1.6 – Функции преобразования типов

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

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 возврата 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*

$$\text{MsgBox}(\text{prompt}[, \text{buttons}][, \text{title}] ),$$

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
A=Worksheets(1).Cells(1,2).Value	Переменной А присваивается значение из ячейки В1 (строка 1, столбец 2) листа 1
Worksheets(1).Cells(4,2).Value= X	В ячейку В4 (строка 4, столбец 2) записывается значение переменной X

*1.3.4 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*

*1.3.5 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*

где <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*.

**1.3.6 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]
```

где *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]
```

где *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.

#### 1.4 Процедуры и функции

В программировании сложный код программы разбивают на подпрограммы (процедуры и функции). Подпрограмма – это группа операторов, выполняющих законченное действие. Основная программа – программа, реализующая основной алгоритм решения задачи и содержащая в себе обращения к подпрограммам (вызов подпрограмм). В точке вызова подпрограммы ей передается управление, а после выполнения всех действий подпрограммы возвращается в основную программу. В подпрограмме могут быть объявлены собственные переменные, а также параметры подпрограммы для обмена значений с основной программой. В VBA существуют два типа подпрограмм: подпрограммы-функции и подпрограммы-процедуры (таблица 1.11). Функция, в отличие от процедуры, возвращает значение через свое имя и может входить в состав выражений.

<Список параметров> в процедурах и функциях отличается от <списка аргументов> (списка фактических параметров) тем, что первый указывается при описании подпрограммы, второй – при ее вызове в основной программе.

<Список параметров> позволяет передать в подпрограмму требуемые значения из вызывающей программы и имеет следующий синтаксис:

```
[ByRef|ByVal] <имя параметра1> [As <Тип>], [ByRef|ByVal]
<имя параметра2> [As <Тип>], ...
```

Таблица 1.11 – Структура процедуры и функции

Подпрограмма-процедура	Подпрограмма-функция
Описание	
Sub <имя процедуры> ([<список параметров>]) <операторы> [Exit Sub] <операторы> End Sub	Function <имя функции> [(<список параметров>)] [As <тип функции>] <операторы> [Exit Function] <операторы> <имя функции> = <выражение> End Function
Вызов в основной программе	
1) <имя процедуры>(<список аргументов>); 2) Call <имя процедуры> [(<список аргументов>)]	<имя функции>(<список аргументов>)

Ключевое слово *ByVal* (передача по значению) используется в тех случаях, когда необходимо, чтобы изменение параметров внутри процедуры не приводило к изменению соответствующих аргументов в основной программе. Использование *ByRef* (передача по ссылке) или, если ключевое слово опущено, означает, что при изменении параметров внутри процедуры происходит изменение соответствующих параметров в основной программе.

<Список аргументов> перечисляется через запятую. Количество и типы параметров и аргументов должны соответствовать. Аргументы, соответствующие параметрам с ключевым словом *ByRef* (или по умолчанию), должны быть переменными.

<Тип> позволяет явно задать тип передаваемых значений. Если тип опущен, то по умолчанию принимает значение *Variant*.

Для выхода из подпрограммы и возврата в основную программу, опуская оставшиеся операторы, используется *Exit Sub* (в процедурах) и *Exit Function* (в функциях).

### 1.5 Обработка строк

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

Строковая константа – последовательность символов, заключенных в кавычки. Например: “это строковая константа”, “256”. Две следующие друг за другом кавычки (“”) обозначают “пустую строку”, т. е. строку нулевой длины.

Строковая переменная описывается в разделе описания переменных:

*Dim <идентификатор> As String*

Например:

*Dim Name As String*

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

**Пример** – “Город”&” Могилев”. В результате получится строка: “Город Могилев”. Длина результирующей строки не должна превышать 255 символов.

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

Для работы со строками могут использоваться следующие строковые функции.

**Left(S, N)** – выделяет из строки S подстроку длиной N символов, начиная с левого края строки.

**Len(S)** – определяет текущую длину строки S. Результат – значение целого типа.

**InStr(N, S1, S2)** – обнаруживает первое появление в строке S2 подстроки S1. Поиск начинается с позиции N, этот аргумент необязателен. Результат – целое число, равное номеру позиции, где находится первый символ подстроки S1. Если в S2 подстроки S1 не обнаружено, то результат равен 0.

**Mid(S, Poz, N)** – формирует новую строку длиной N из символов строки S, взятых подряд, начиная с позиции Poz.

**Right(S, N)** – формирует строку длиной N символов, взятых подряд из строки S, начиная с правого края.

## **1.6 Интерфейс редактора VBA**

Редактор VBA вызывается командой Сервис/Макрос/Редактор Visual Basic (Excel 2003) или кнопкой Visual Basic на вкладке Разработчик (Excel 2007...). Перейти из редактора VBA в рабочую книгу можно нажатием кнопки Вид MS Excel.

Программа на языке VBA размещается в модуле, который добавляется к файлу Excel, и сохраняется вместе с ним.

Интерфейс редактора VBA состоит из следующих основных элементов: окно проекта, окно свойств, окно редактирования кода, окна форм, меню и панели инструментов (рисунок 1.1).

Окно проекта содержит иерархическую структуру файлов форм и модулей текущего проекта. В проекте автоматически создаются модули для каждого рабочего листа и для всей книги. Кроме того, модули создаются для каждой пользовательской формы, макросов и классов.

В окне проекта выводится проект всех открытых рабочих книг. Это позволяет легко копировать формы и коды из одного проекта в другой, что

ускоряет процесс создания новых приложений.

В окне свойств перечисляются основные установки свойств выбранной формы или элемента управления. Используя это окно, можно просматривать свойства и изменять их установки. Для просмотра свойств выбранного объекта достаточно либо щелкнуть кнопку Окно свойств, либо выбрать команду Вид / Окно свойств (View/Property Window).

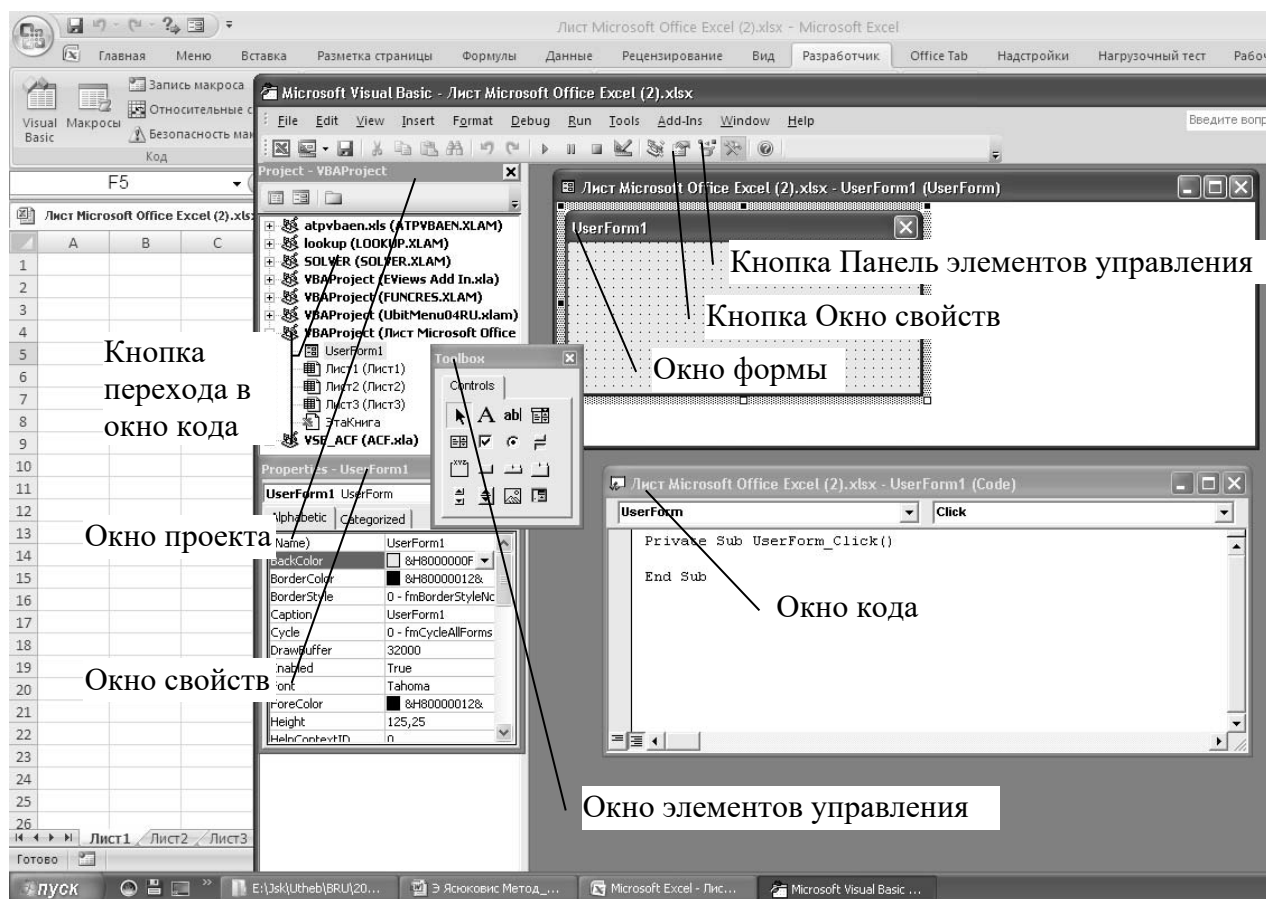


Рисунок 1.1 – Окно VBA


Код программы записывается в окне кода, которое открывается двойным щелчком мыши на объекте в окне проекта.


Окно модуля открывается командой Вставка / Модуль (Insert / Module).

*1.6.1 Формы и элементы управления.* Форма (пользовательская форма) представляет собой диалоговое окно, в котором можно размещать различные элементы управления. В приложении может быть как одна, так и несколько форм. Новая форма добавляется в проект выбором команды Insert / UserForm.

В VBA имеется обширный набор встроенных элементов управления. Используя этот набор и редактор форм, нетрудно создать любой пользовательский интерфейс, который будет удовлетворять всем требованиям, предъявляемым к интерфейсу в среде Windows. Элементы управления являются объектами. Как любые объекты, они обладают свойствами, методами и событиями. Элементы управления создаются при помощи Панели элементов, которая



отображается на экране, либо выбором команды Вид (View) / Панель элементов (Toolbox), либо нажатием кнопки .

Для создания элементов управления служат все кнопки панели инструментов, за исключением кнопки Выбор объекта . Щелкнув по кнопке Выбор объекта, можно выбрать уже созданный в форме элемент управления для последующего его редактирования.

Список основных элементов управления и соответствующих кнопок панели элементов приведен в таблице 1.12.

Для размещения элемента управления на форму необходимо выбрать соответствующую кнопку на панели элементов и с помощью мыши перетащить рамку элемента управления в нужное место. После этого элемент управления можно перемещать, изменять его размеры, копировать в буфер обмена, вставлять из буфера обмена и удалять из формы. Основные общие свойства элементов управления приведены в таблице 1.13.

Таблица 1.12 – Основные элементы управления

Элемент управления	Имя	Кнопка	Элемент управления	Имя	Кнопка
Поле	TextBox		Переключатель	OptionButton	
Надпись	Label		Флажок	CheckBox	
Кнопка	CommandButton		Выключатель	ToggleButton	
Список	ListBox		Рамка	Frame	
Поле со списком	ComboBox		Рисунок	Image	
Полоса прокрутки	ScrolBar		Набор страниц	MultiPage	
Счетчик	SpinButton		Набор вкладок	TabStrip	

Таблица 1.13 – Основные общие свойства элементов управления

Свойство	Описание
Caption	Надпись, отображаемая при элементе управления
AutoSize	Допустимые значения: True (устанавливает режим автоматического изменения размеров элемента управления так, чтобы на нем полностью помещался текст, присвоенный свойству Caption) и False (в противном случае)
Visible	Допустимые значения: True (элемент управления отображается во время выполнения программы) и False (в противном случае)
Enabled	Допустимые значения: True (пользователь вручную может управлять элементом управления) и False (в противном случае)
Height и Width	Устанавливают геометрические размеры объекта (высоту и ширину)
Left и Top	Устанавливают координаты верхнего левого угла элемента управления, определяющие его местоположение в форме

Окончание таблицы 1.13

Свойство	Описание
ControlTipText	Устанавливает текст в окне всплывающей подсказки, связанной с элементом управления
BackColor, ForeColor и BorderColor	Устанавливают цвет заднего и переднего плана элемента управления, и его границы
BackColor	Устанавливает тип заднего фона
BorderStyle	Устанавливает тип границы. Допустимые значения: fmBorderStyleSingle (граница в виде контура); fmBorderStyleNone (граница невидима)
SpecialEffect	Устанавливает тип границы. Отличается от свойства BorderStyle тем, что позволяет установить несколько типов, но одного цвета. BorderStyle позволяет установить только один тип, но различных цветов
Picture (создание картинки)	Внедряет картинку на элемент управления. Например, на поверхности кнопки картинка отображается с помощью следующей инструкции: CommandButton1 Picture = LoadPicture("c:\mydoc\Круг.bmp") Функция LoadPicture(ПолноеИмяФайла) считывает графическое изображение. Аргумент ПолноеИмяФайла указывает полное имя графического файла
Picture (удаление картинки)	После того как картинка создана на элементе управления, иногда возникает необходимость ее удалить. Это легко достигается присвоением свойству Picture значения LoadPicture("") CommandButton1.Picture = LoadPicture("")

## 2 Методические указания к выполнению лабораторных работ

В данном разделе приводятся методические указания для выполнения девяти лабораторных работ. Первые шесть работ предназначены для изучения основ программирования, поэтому в них пользовательские формы не используются. Три последние лабораторные работы выполняются с применением пользовательских форм и предполагают разработку на их основе пользовательских интерфейсов задач.

Перед выполнением каждой лабораторной работы необходимо ознакомиться с теоретическими сведениями, содержащимися в данных методических рекомендациях. Для этого можно использовать также конспект лекций или рекомендуемую литературу.

Каждая лабораторная работа выполняется в следующем порядке: постановка задачи; разработка схемы алгоритма; разработка программного модуля (процедуры или функции); отладка программного модуля; решение поставленной задачи; анализ результатов; оформление отчета.

По каждой лабораторной работе должен быть оформлен отчет, содержащий:

- постановку задачи;
- условие задачи;

- схему алгоритма;
- текст отлаженной задачи;
- результаты;
- выводы по работе.

### 2.1 Лабораторная работа № 1. Программирование с использованием объектов Excel. Линейная программа

*Цель работы:* изучение среды программирования VBA, методов ввода с листа Excel и вывода на лист Excel; разработка и отладка простейшей программы.

**Постановка задачи.** Составить процедуру (линейную программу) для вычисления значения выражения, приведенного в таблице 2.1. Исходные данные (переменные  $a = 2,27$ ,  $b = 1,45$ ,  $x = 0,75$ ) ввести с ячеек соответственно B1, B2, B3 первого листа Excel. В ячейках A1, A2, A3, A4 ввести комментарий: “a= “, “b= “, “x= “, “y= “.

Результат вычисления, переменную  $y$ , вывести в ячейку B4.

Таблица 2.1 – Варианты заданий для выполнения лабораторной работы № 1

Номер варианта	Формула	Номер варианта	Формула
1	$y = \frac{a+b \cdot x}{a-x} \sin^3(x/a - 0,2)^2$	9	$y = \sqrt[4]{b \cdot x^{1,75}} + \frac{b-x^2}{a+x^2/b} \operatorname{ctg}(x^2 - 0,2)^4$
2	$y = \frac{a \cdot b/x}{a+b \cdot x} \cos^2(x/a - 0,15)^{2,4}$	10	$y = \frac{a \cdot b + x}{a - b \cdot x} \operatorname{tg}^2(x/b \cdot a - 0,2)^{2,5}$
3	$y = \sqrt[3]{a \cdot x} + \frac{a \cdot x}{b+x} \cos^2(x/1,5)^3$	11	$y = \frac{a/x}{b-x^4} \cos^3(x/1,4)^3 + \sqrt[4]{b/x}$
4	$y = \sqrt[4]{a/x} + \frac{a-x^2}{b+x^3} \operatorname{tg}^2(x+0,25)^3$	12	$y = \frac{a+x^2}{b+x^2} \operatorname{tg}^2(x+0,2)^3 - \sqrt[4]{b/x}$
5	$y = \sqrt[4]{a \cdot x^{1,75}} + \frac{a+x^3}{b-x^2} \operatorname{tg}^3(x^2 - 0,2)^2$	13	$y = \frac{a-x^3}{b+x^2} \operatorname{ctg}^2(x^2 - 0,25) \cdot 3 - \sqrt[3]{a \cdot b/x}$
6	$y = \frac{a-b/x}{a+b \cdot x} \operatorname{tg}^2(x/b - 0,12)^{2,5}$	14	$y = \operatorname{tg}^2(x/a^3 - 0,125)^3 + \frac{a \cdot x - b/x}{(a+b) \cdot x}$
7	$y = \sqrt[4]{a \cdot x^{1,7}} + \frac{b-x^2}{a+x^2/b} \operatorname{ctg}(x^2 - 0,2)^4$	15	$y = \frac{b-x^2}{a+x^2/b} \operatorname{ctg}^2(x^3 - 0,1)^4 - \sqrt[3]{a/b \cdot x^{1,2}}$
8	$y = \frac{a-b \cdot x}{a-x^2} \sin^2(x^3/a + 0,2)^{2,4}$	16	$y = \sin^3(x^{1,2}/a + 0,2)^2 + \frac{a+b/x}{a/b - x^2}$

#### Контрольные вопросы

- 1 Как вызвать среду программирования VBA?
- 2 Как вызвать окно кода?
- 3 Как создать процедуру в среде VBA?

- 4 Как создать функцию в среде VBA?
  - 5 Как организовать ввод исходных данных в процедуре?
  - 6 В каком порядке выполняются операции в выражениях VBA?
  - 7 Как возвести функцию в степень в выражении?
  - 8 Для чего используются круглые скобки в выражениях?
- Пример программы к лабораторной работе № 1

Исходные данные (Лист 1)

	A	B	C
1	a=	2,27	
2	b=	1,45	
3	x=	0,75	
4	y=	1,723	
5			

Процедура Листа 1 на VBA

```
Public Sub Lb1()
    a = Cells(1, 2):    b = Cells(2, 2)
    x = Cells(3, 2)
    y = Sin((x ^ 1.2 / a + 0.2) ^ 2) ^ 3 _
        + (a + b / x) / (a / x - x ^ 2)
    Worksheets(1).Range("B4") = y
End Sub
```

## 2.2 Лабораторная работа № 2. Разветвляющаяся программа

*Цель работы:* изучение условного оператора If, функций ввода InputBox и вывода MsgBox; разработка и отладка разветвляющейся программы.

**Постановка задачи.** Составить процедуру с разветвлением для вычисления значения  $y$  на основе одного из выражений в зависимости от заданного условия.

Варианты заданий приведены в таблице 2.2. Исходные данные (переменные  $a = 1,25$ ;  $x = 0,25$ ) ввести с помощью функции InputBox в виде

$$x = \text{InputBox}(\text{"Ввод переменной } x", \text{"Окно ввода, 27})$$

Переменную  $b$  вычислить по формуле  $b = 1,5 \times a$  и вывести с помощью функции MsgBox(prompt[,buttons][,title]) например, в виде

$$\text{Call MsgBox}(b)$$

Результат вычисления, переменную  $y$ , вывести в ячейку A3 в виде “ $y = \text{XX.XXX}$ ”, где XX.XXX – вычисленное значение переменной  $y$ .

В ячейках A1, A2 на Лист 1 ввести комментарий: “ $a =$ “, “ $x =$ “, а в ячейки B1, B2 – численные значения переменных  $a, x$ .

Таблица 2.2 – Варианты заданий для выполнения лабораторной работы № 2

Номер варианта	Задание	Номер варианта	Задание
1	$y = \begin{cases} x^3 + a \cdot x^2 - e^{2 \cdot x}, & \text{если } x < 2,5 \\ \frac{a}{x^2 + b} \ln x, & \text{если } x \geq 2,5 \end{cases}$	9	$y = \begin{cases} a \cdot x^2 + b \cdot x^2 - e^x, & \text{если } x < 1,5 \\ \frac{a/b}{x^2 + a} \ln x^2, & \text{если } x \geq 1,5 \end{cases}$
2	$y = \begin{cases} a \cdot x^2 + b \cdot x - e^{x/2}, & \text{если } x < 0,25 \\ \frac{b}{x^2 + a} \ln^2 x, & \text{если } x \geq 0,25 \end{cases}$	10	$y = \begin{cases} b \cdot x^3 + a \cdot x^3 - e^x, & \text{если } x < 0,5 \\ \frac{b}{x^2 + a} \ln^2 x, & \text{если } x \geq 0,5 \end{cases}$
3	$y = \begin{cases} \frac{a \cdot x^2}{2} + b \cdot x + e^x, & \text{если } x < 0,5 \\ \frac{b}{x + a^2} \ln x, & \text{если } x \geq 0,5 \end{cases}$	11	$y = \begin{cases} b \cdot x^3 + a \cdot x^3 - e^x, & \text{если } x < 0,5 \\ \frac{b}{x^2 + a} \ln^2 x, & \text{если } x \geq 0,5 \end{cases}$
4	$y = \begin{cases} \frac{a - x^2}{b} + x + e^x, & \text{если } x < 0,75 \\ \frac{b - x}{x + a^2} \ln x^2, & \text{если } x \geq 0,75 \end{cases}$	12	$y = \begin{cases} \frac{b}{x} \cdot x^{2,5} + x^2 - e^{x+2}, & \text{если } x < 0,1 \\ \frac{b}{x^2 + a} \ln^2 x, & \text{если } x \geq 0,1 \end{cases}$
5	$y = \begin{cases} \frac{-x^2}{a \cdot b} + x + e^{2 \cdot x}, & \text{если } x < 0,72 \\ \frac{b + x}{x - a} \ln^3 x, & \text{если } x \geq 0,72 \end{cases}$	13	$y = \begin{cases} \frac{b}{x^2} + x - e^{2 \cdot x}, & \text{если } x < 0,3 \\ \frac{b}{x^2 + a} \ln^2 x, & \text{если } x \geq 0,3 \end{cases}$
6	$y = \begin{cases} \frac{a \cdot b}{x^2} + x + e^{2 \cdot x}, & \text{если } x < 0,7 \\ \frac{b - x}{x + a} \ln^3 x, & \text{если } x \geq 0,7 \end{cases}$	14	$y = \begin{cases} \frac{-a}{x^2} - x - e^{2 \cdot x}, & \text{если } x < 0,8 \\ \frac{2 \cdot a}{x^2 + b} \ln^3 x, & \text{если } x \geq 0,8 \end{cases}$
7	$y = \begin{cases} x^3 - b \cdot x^2 - e^x, & \text{если } x < 2,4 \\ \frac{a - x}{x^2 + b} \ln x, & \text{если } x \geq 2,4 \end{cases}$	15	$y = \begin{cases} \frac{-b}{x^2} + x - e^{2,4 \cdot x}, & \text{если } x < 0,6 \\ (a - 1,2) \ln^2 x, & \text{если } x \geq 0,6 \end{cases}$
8	$y = \begin{cases} x^3 + a \cdot x - e^x, & \text{если } x < 2,45 \\ \frac{a + x}{x^2 - b^2} \ln^3 x, & \text{если } x \geq 2,45 \end{cases}$	16	$y = \begin{cases} \frac{a}{(x^2 + 1)} + x - e^{2,5 \cdot x}, & \text{если } x < 0,3 \\ (a + 1) \ln^2 x^3, & \text{если } x \geq 0,3 \end{cases}$

### Контрольные вопросы

- 1 Какие формы записи условного оператора Вы знаете?
- 2 Для чего используется функция *InputBox* в языке VBA?
- 3 Какова форма записи функции вывода *MsgBox*?
- 4 Какие форма записи оператора *If* Вы знаете?

- 5 В каких случаях условный оператор можно записать в одну строку?
- 6 Чем заканчивается область действия условного оператора?
- 7 Какой синтаксис имеет оператор *If*?
- 8 В каких случаях элемент *End If* в операторе *If* может не использоваться?

### 2.3 Лабораторная работа № 3. Оператор выбора *Select Case*

*Цель работы:* изучение оператора выбора *Select Case*; разработка и отладка процедуры с выбором вариантов.

**Постановка задачи.** Составить процедуру с разветвлением для выбора одного из значений в зависимости от значения переменной  $k$ , которое определяется на основе генерируемого с помощью функции *Rnd* псевдослучайного числа:

$$k = \text{Int}(m * \text{Rnd}) + 1,$$

где  $\text{Int}(x)$  – функция выделения целой части числа (см. таблицу 1.4);

$m$  – количество возможных значений  $k$ .

Для определения листа Excel для вывода результата использовать инструкцию *With...End With*.

Варианты заданий к лабораторной работе приведены в таблице 2.3.

Таблица 2.3 – Варианты заданий для выполнения лабораторной работы № 3

Номер варианта	$m$	Задание
1	4	Определить пору года: 1 – зима; 2 – весна и т. д.
2	12	Определить название месяца года: 1 – январь; 2 – февраль и т. д.
3	3	Определить месяц зимы: 1 – декабрь; 2 – январь; 3 – февраль
4	3	Определить месяц весны: 1 – март; 2 – апрель; 3 – май
5	10	Определить оценку прописью: 1 – один; 2 – два и т. д.
6	3	Определить месяц лета: 1 – июнь; 2 – июль; 3 – август
7	3	Определить месяц осени: 1 – сентябрь; 2 – октябрь; 3 – ноябрь
8	7	Определить день недели: 1 – понедельник; 2 – вторник и т. д.
9	6	Определить областной город Беларуси: 1 – Минск; 2 – Брест; 3 – Могилев; 4 – Гомель; 5 – Витебск; 6 – Гродно
10	8	Определить название цвета: 1 – роза; 2 – хризантема; 3 – тюльпан; 4 – подснежник; 5 – одуванчик; 6 – пион; 7 – астра; 8 – гладиолус
11	9	Определить название дерева: 1 – береза; 2 – ясень; 3 – дуб; 4 – клен; 5 – ольха; 6 – ель; 7 – липа; 8 – тополь; 9 – осина
12	4	Определить тип принтера: 1 – лазерный; 2 – струйный; 3 – матричный; 4 – барабанный
13	6	Определить фамилию писателя: 1 – Л. Толстой; 2 – А. Пушкин; 3 – Н. Островский; 4 – О. Генри; 5 – С. Есенин; 6 – И. Тургенев

## Окончание таблицы 2.3

Номер варианта	$m$	Задание
14	7	Определить фамилию математика: 1 – Н. Винер; 2 – Р. Фишер; 3 – М. Бернадский; 4 – Д. Нейман; 5 – С. Заремба; 6 – К. Пирсон; 7 – А. Ляпунов
15	7	Определить название предметов: 1 – физика; 2 – астрономия; 3 – алгебра; 4 – химия; 5 – география; 6 – биология; 7 – геометрия
16	8	Определить названия государств: 1 – Россия; 2 – Беларусь; 3 – Украина; 4 – Эстония; 5 – Швеция; 6 – Китай; 7 – Ливан; 8 – Польша

**Контрольные вопросы**

- 1 Каков принцип работы инструкции Select Case?
- 2 Как организовать ввод исходных данных с листа Excel?
- 3 Назначение инструкции With...End With.
- 4 Как организовать вывод в ячейку таблицы Excel текста и численного значения переменной?
- 5 Какие средства отладки кода в среде Excel Вы знаете?

**2.4 Лабораторная работа № 4. Массивы и циклы For...Next**

*Цель работы:* изучение основных принципов работы с массивами, оператора For...next, разработка и отладка циклической программы.

**Постановка задачи.** В работе выполняется задание 1, либо задание 2, либо задание 3 по указанию преподавателя.

**Задание 1**

Составить процедуру Sub() формирования массива X(25) вещественного типа и вычисления параметра, указанного в таблице 2.4.

Таблица 2.4 – 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 двухмерный массив y, первый столбец которого содержит положительные элементы массива x, 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

## Окончание таблицы 2.4

Номер варианта	Задание
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 $y$
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

*Примечание* – Величина  $r$  – это формируемое функцией  $Rnd$  псевдослучайное число в интервале  $0...1$ .

**Задание 2**

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

**Задание 3**

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

**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.

### 2.5 Лабораторная работа № 5. Циклы *While...Wend* и *Do...Loop*

*Цель работы:* изучение операторов цикла, принципов использования отладочной печати Debug Print, разработка и отладка циклической программы.

**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 2.5.

Таблица 2.5 – Варианты заданий для выполнения лабораторной работы № 5

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

### **Контрольные вопросы**

- 1 Прокомментируйте назначение оператора *Do*.
- 2 Какие операторы цикла Вы знаете?
- 3 Как работает оператор цикла *Do* с предусловием?
- 4 Для чего используется оператор *For*?
- 5 Как работает оператор цикла с постусловием?
- 6 Для чего используется оператор отладочной печати?
- 7 В какой объект выводятся данные отладочной печати?
- 8 Сформулируйте условия выхода из цикла *Do*.

### **2.6 Лабораторная работа № 6. Численные методы**

*Цель работы:* изучение численного метода касательных решения нелинейного уравнения, алгоритма решения нелинейного уравнения, а также процедур и функций VBA; разработка и отладка программы решения нелинейного уравнения.

**Постановка задачи.** В лабораторной работе требуется разработать алгоритм и программу вычисления корня нелинейного уравнения, используя численный метод деления отрезка пополам или метод касательных (по указанию преподавателя).

Таблица 2.5 – Варианты заданий для выполнения лабораторной работы № 6

Номер варианта	Задание	Номер варианта	Задание
1	$x - \sin x = 0,25$	9	$5x - 8 \ln x = 8$
2	$\sqrt{x} - \cos x = 0$	10	$x \ln(x + 1) = 0,5$
3	$3x - \cos x - 1 = 0$	11	$\operatorname{tg} x = x$
4	$x + \lg x = 0,5$	12	$e^{-x} = \sin x$
5	$\operatorname{tg} x = \cos x - 0,1$	13	$2 + \ln x = 1/x$
6	$\operatorname{ctg} x = x$	14	$\ln x = \sin x$
7	$x \ln(x + 1) - 0,3 = 0$	15	$\ln x = 1/x_1$
8	$(x + 3)\cos x = 1$	16	$e^{-x} = x$

### **Контрольные вопросы**

- 1 Как решить графически задачу поиска корня нелинейного уравнения?
- 2 В чем состоит основная идея решения нелинейного уравнения методом деления отрезка пополам?
- 3 В чем состоит основная идея решения нелинейного уравнения методом касательных?
- 4 Для чего используется допустимая погрешность вычисления корня?

5 Какие операторы используются в задаче поиска корней нелинейного уравнения?

6 Какие численные методы решения обыкновенных дифференциальных уравнений Вы знаете?

## **2.7 Лабораторная работа № 7. Пользовательские формы. Элементы управления VBA: Label, TextBox, ComboBox, CommandButton**

*Цель работы:* изучение элементов управления: Label, TextBox, ComboBox, CommandButton; разработка интерфейса задачи решения треугольника на основе объекта UserForm, а также приложения решения треугольника.

**Постановка задачи.** В лабораторной работе требуется:

– построить математическую модель задачи: формулы для вычисления высоты, периметра и площади треугольника;

– организовать ввод исходных данных – длин сторон треугольника. Причем длина  $a$  выбирается из списка ComboBox,  $b$  и  $c$  – вводятся с помощью элемента TextBox;

– организовать проверку достоверности исходных данных и, при необходимости, вывода соответствующего сообщения с помощью функции MsgBox;

\_ запуск приложения должен производиться кнопкой **Start** элемента CommandButton1, а завершение – кнопкой **Stop** элемента CommandButton2;

– интерфейс задачи должен иметь вид, представленный на рисунке 2.1.



Рисунок 2.1 – Окно интерфейса задачи

### **Контрольные вопросы**

- 1 Как вызвать пользовательскую форму в среде VBA?
- 2 Как переместить на форму необходимые для организации интерфейса задачи элементы управления?
- 3 Что такое событие и свойство объекта?
- 4 Как построить процедуру для кнопки Stop?

- 5 Как прочитать численное значение из элемента TextBox?
- 6 Как подготовить к работе элемент управления ComboBox?

## 2.8 Лабораторная работа № 8. Элемент управления Переключатель

*Цель работы:* изучение свойства элемента `OptionButton` и использование его для решения прикладной задачи.

**Постановка задачи.** Разработать программу, выполняющую одну из четырех арифметических операций (сложение, вычитание, умножение, деление) над двумя числами по выбору пользователя.

Для выбора каждой операции использовать элемент управления Переключатель – `OptionButton`.

Для проверки выполняемой операции можно использовать следующий оператор кода:

```
If OptionButton1.Value = True Then ...
```

Форма интерфейса задачи может иметь вид, представленный на рисунке 2.2.

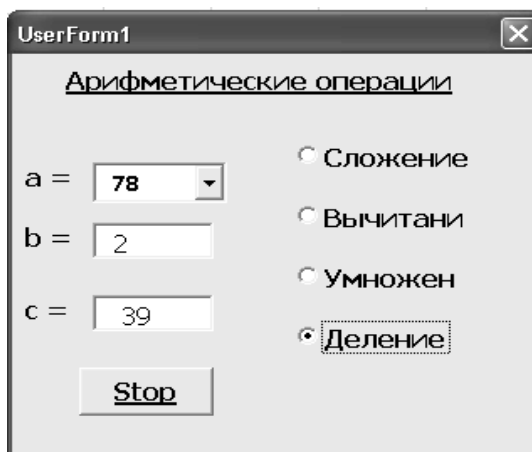


Рисунок 2.2 – Форма задачи

**Постановка задачи.** Создать пользовательскую форму и написать программу Калькулятор. `UserForm` задачи должна содержать: один элемент `ComboBox` для выбора из списка первого числа, один элемент `TextBox` для ввода второго числа, один элемент `TextBox` для вывода результата, четыре элемента `OptionButton` для выбора операций арифметических действий (+, -, \*, /), элемент `CommandButton` для завершения программы – кнопку `Stop`.

### Контрольные вопросы

- 1 Как работает элемент `OptionButton` на форме?
- 2 Как построить процедуру выполнения арифметического действия над двумя переменными?

3 Как организовать вывод результатов выполнения арифметических действий?

4 Как организовать завершение работы VBA приложения?

5 Как изменить название пользовательской формы?

## 2.9 Лабораторная работа № 9. Обработка строк

*Цель работы:* изучение операций с символьными (Chr) и строковыми данными (String).

**Постановка задачи.** Задан текстовый фрагмент. Необходимо выполнить с ним действия, указанные в таблице 2.7, согласно заданному варианту.

Для выполнения вариантов заданий данной лабораторной работы следует использовать следующий текст:

«Объектно-ориентированное программирование предоставляет возможность применять при разработке приложений такие объекты как рабочий лист, диапазон ячеек, пользовательская форма, таблица, диаграмма – в Excel; документ, шрифт, область в документе, стиль в документе, таблица – в Word».

Варианты заданий для выполнения работы приведены в таблице 2.7.

Таблица 2.7 – Варианты заданий для выполнения лабораторной работы № 9

Номер варианта	Задание
1	Задан текст. Проверить, какая из букв «а» или «е» встречается чаще
2	Задан текст. Вывести номера позиций, на которых встречается буква «к»
3	Определить, сколько в тексте слов заканчивается на букву «о»
4	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»
5	Определить, сколько раз в тексте встречается введенное слово
6	Определить, какой процент слов в тексте содержит хотя бы одну букву «d»
7	Определить, какой процент слов в тексте содержит удвоенную согласную
8	Напечатать те слова из текста, в которых нет удвоенной буквы
9	Удалить из текста все удвоенные буквы
10	Из заданного текста удалить последовательность символов между двумя двоеточиями (хотя бы одна пара двоеточий имеется)
11	Перед каждой цифрой в тексте вставить знак «+»
12	Напечатать самое длинное слово из заданного текста
13	Определить, является ли введенное 6-значное число числом-палиндромом. Числа-палиндромы – это такие числа, которые читаются одинаково как справа налево, так и слева направо, т. е. 654456, 112211 и т. д.
14	Определить является ли одно слово вхождением другого, т. е. например слово «грамм» является вхождением слова «программа»
15	Напечатать удвоенные буквы в строке и их позиции
16	Заменить в тексте букву «о» на цифру «0», а букву «з» на цифру «3». Вывести количество таких замен

Результат выполнения операций с текстом вывести в окно ListBox на UserForm.

### **Пример решения варианта 1**

Для решения задачи в ячейку первого листа таблицы Excel введем текст и построим форму (рисунок 2.3), на которой разместим элементы управления: Label1 с текстом «Проверка строки», элемент TextBox1 для вывода результата и элементы CommandButton1 со свойством Caption = «Start» и CommandButton2 со свойством Caption = «Stop» для запуска процедуры на выполнения и ее завершения.

На рисунке 2.3 приведена также форма UserForm с результатом выполнения.

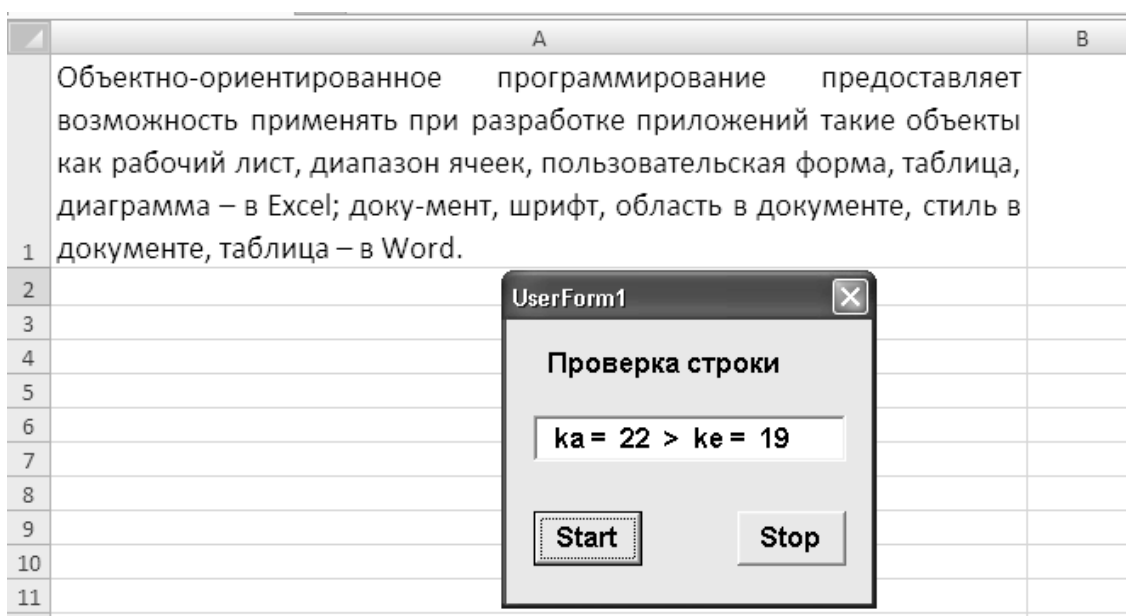


Рисунок 2.3 – Текстовый фрагмент и проектируемая пользовательская форма

Текст программы состоит из процедур CommandButton1\_Click() и CommandButton2\_Click() и имеет следующий вид:

```
Private Sub CommandButton1_Click()
    s = Cells(1, 1)
    ka = 0: ke = 0
    For i = 1 To Len(s)
        sa = Mid(s, i, 1)
        If sa = "a" Then ka = ka + 1
        If sa = "e" Then ke = ke + 1
    Next i
    If ka > ke Then
        TextBox1.Text = "ka = " & Str(ka) & " > ke = " & Str(ke)
    Else
        TextBox1.Text = "ke = " & Str(ke) & " > ka = " & Str(ka)
    End If
End Sub
```

*End If*  
*End Sub*

*Private Sub CommandButton2\_Click()*  
*Unload Me*  
*End Sub*

### ***Контрольные вопросы***

- 1 Какие стандартные функции для работы с текстовыми фрагментами Вы знаете?
- 2 Какая функция VBA используется для определения длины (количества символов) строки?
- 3 Какая операция используется для конкатенации (объединения) строк текста?
- 4 Какая функция используется для преобразования численного значения в строковое?
- 5 Как построить программный блок вычисления количества заданных символов в строке?

### **Список литературы**

- 1 **Лебедев, В. М.** Программирование на VBA в MS Excel : учеб. пособие / В. М. Лебедев. – 3-е изд., испр. и доп. – М. : Юрайт, 2023. – 312 с.
- 2 **Лукин, С. Н.** Visual Basic 6.0. Самоучитель для начинающих / С. Н. Лукин. – М. : Диалог; МИФИ, 2001. – 544 с.
- 3 **Сергеев, В.** Visual Basic 6.0 : пер. с англ. / В. Сергеев. – СПб. : БХВ-Петербург, 2004. – 992 с.: ил.
- 4 **Кильдишов, В. Д.** MS Excel и VBA для моделирования различных задач / В. Д. Кильдишов. – М.: СОЛОН-Пресс, 2023. – 258 с.