

МИНИСТЕРСТВО ТРАНСПОРТА И СВЯЗИ УКРАИНЫ
ДОНЕЦКИЙ ИНСТИТУТ ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
УКРАИНСКОЙ ГОСУДАРСТВЕННОЙ АКАДЕМИИ
ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

**Методические указания
к лабораторным работам
Раздел. «ПРОГРАММИРОВАНИЕ В СРЕДЕ
VISUAL BASIC FOR APPLICATION»**

Донецк-2011

Методические указания к лабораторным работам. Раздел. «ПРОГРАММИРОВАНИЕ В СРЕДЕ VISUAL BASIC FOR APPLICATION» / Сост.: С.В. Вакуленко, М.Р. Терованесов, Е.И. Демидова, Е.А. Литвинова.

В данном пособии представлены методические указания к лабораторным работам по теме «ПРОГРАММИРОВАНИЕ В СРЕДЕ VISUAL BASIC FOR APPLICATION». В состав пособия входят примеры решения семи лабораторных работ, которые охватывают основные начальные элементы программирования в среде Visual Basic for Application.

Пособие разработано для студентов дневной и заочной форм обучения. Оно может быть использовано как практическое руководство к решению задач и другими пользователями, желающими самостоятельно постичь основы программирования.

© Донецкий институт железнодорожного транспорта, 2011
© Вакуленко С.В., Терованесов М.Р.,
Демидова Е.И., Литвинова Е.А.

СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА № 1 Тема: «Организация интерфейса для реализации линейных процессов в среде VBA»	5
ЛАБОРАТОРНАЯ РАБОТА № 2 Тема: «Организация интерфейса для реализации разветвляющихся процессов в среде VBA»	12
ЛАБОРАТОРНАЯ РАБОТА № 3 Тема: «Организация интерфейса для реализации циклических процессов в среде VBA. Цикл с предусловием»	20
ЛАБОРАТОРНАЯ РАБОТА № 4 Тема: «Организация интерфейса для реализации циклических процессов в среде VBA. Цикл с постусловием»	28
ЛАБОРАТОРНАЯ РАБОТА № 5 Тема: «Организация интерфейса для реализации циклических процессов в среде VBA. Цикл с блоком итераций»	36
ЛАБОРАТОРНАЯ РАБОТА № 6 Тема: «Организация интерфейса для реализации циклических процессов в среде VBA. Одномерные массивы»	44
ЛАБОРАТОРНАЯ РАБОТА № 7 Тема: «Организация интерфейса для реализации циклических процессов в среде VBA. Двумерные массивы»	52
ПРИЛОЖЕНИЕ I	60

ПОЯСНЕНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

1. Для лабораторных работ № 1-7 необходимо составить алгоритм в виде блок-схемы и программу по заданному условию. Программу необходимо:
 - а) набрать на компьютере в среде Visual Basic;
 - б) отладить;
 - в) сделать контрольный просчет;
 - г) распечатать результаты вычисления программы и ее код;
 - д) распечатать данные с листов Microsoft Excel (если необходимо);
 - е) распечатанные результаты и код программы необходимо вклеить в тетрадь.

Номер варианта студент берет по номеру в списке журнала группы.

ЛАБОРАТОРНАЯ РАБОТА № 1

Тема: «Организация интерфейса для реализации линейных процессов в среде VBA»

Цель: *Получение навыков составления программ линейных вычислительных процессов.*

Пример решения задания

Задание

Вычислить X , Y . Исходные данные: $A = 35$; $B = -9$; $C = -4$.

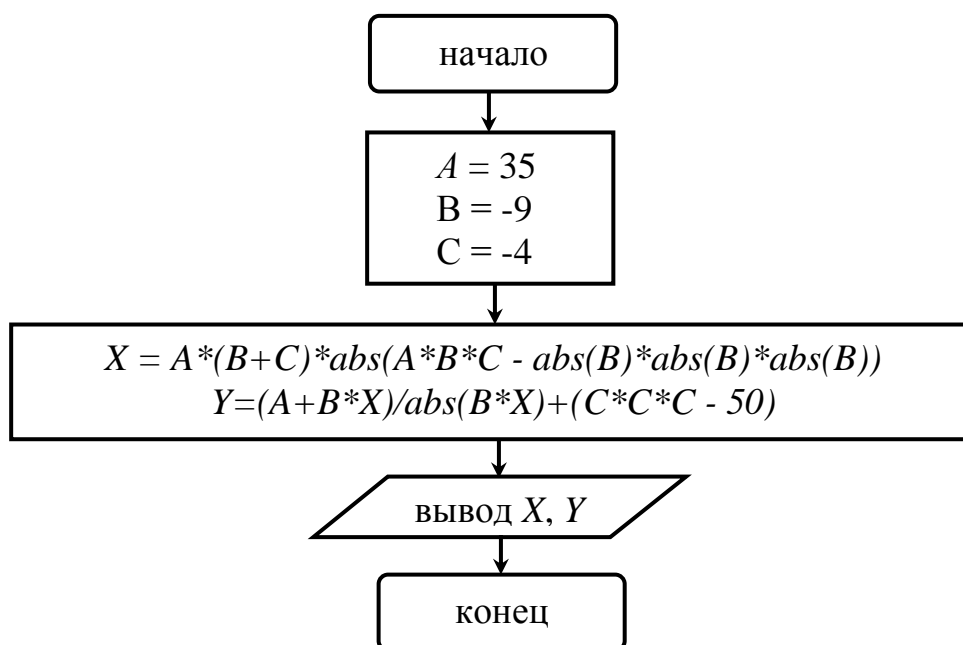
$$X = A(B + C) \left| ABC - |B|^3 \right|$$

$$Y = \frac{A + Bx}{|Bx|} + (C^3 - 50)$$

Вывести значения X , Y .

Решение поставленной задачи

ЭТАП 1. Составление блок-схемы



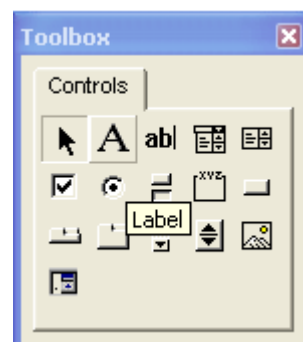
Как видно из блок-схемы, выражения расчета X , Y в блок-схеме претерпели изменения по сравнению с условием. Соответствие записи условия и блок-схемы приведены в ПРИЛОЖЕНИИ I.

ЭТАП 2. Определение элементов управления ввода-вывода

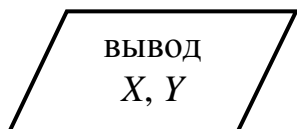
На данном этапе необходимо определиться с элементами управления, которые будут использоваться для ввода и вывода данных. Добавление элементов происходит с панели инструментов *Toolbox* (в случае отсутствия данной панели на экране необходимо: *Insert* → *Toolbox*)

ИСХОДНЫЕ ДАННЫЕ

В данном случае нет неизвестных данных, требующих ввода, поэтому для указания значений известных переменных *A*, *B*, *C* можно использовать элемент управления **Label**. Для данного элемента управления свойства можно не менять, т. к. он не будет использоваться в программе.



РЕЗУЛЬТАТЫ

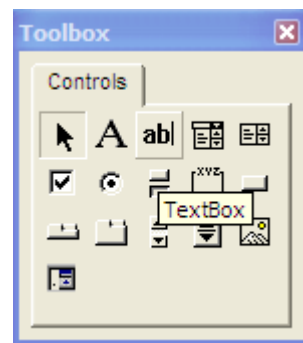


Наличие в блок-схеме блока вывода данных *X*, *Y* требует добавления в форму двух элементов управления типа **TextBox**.

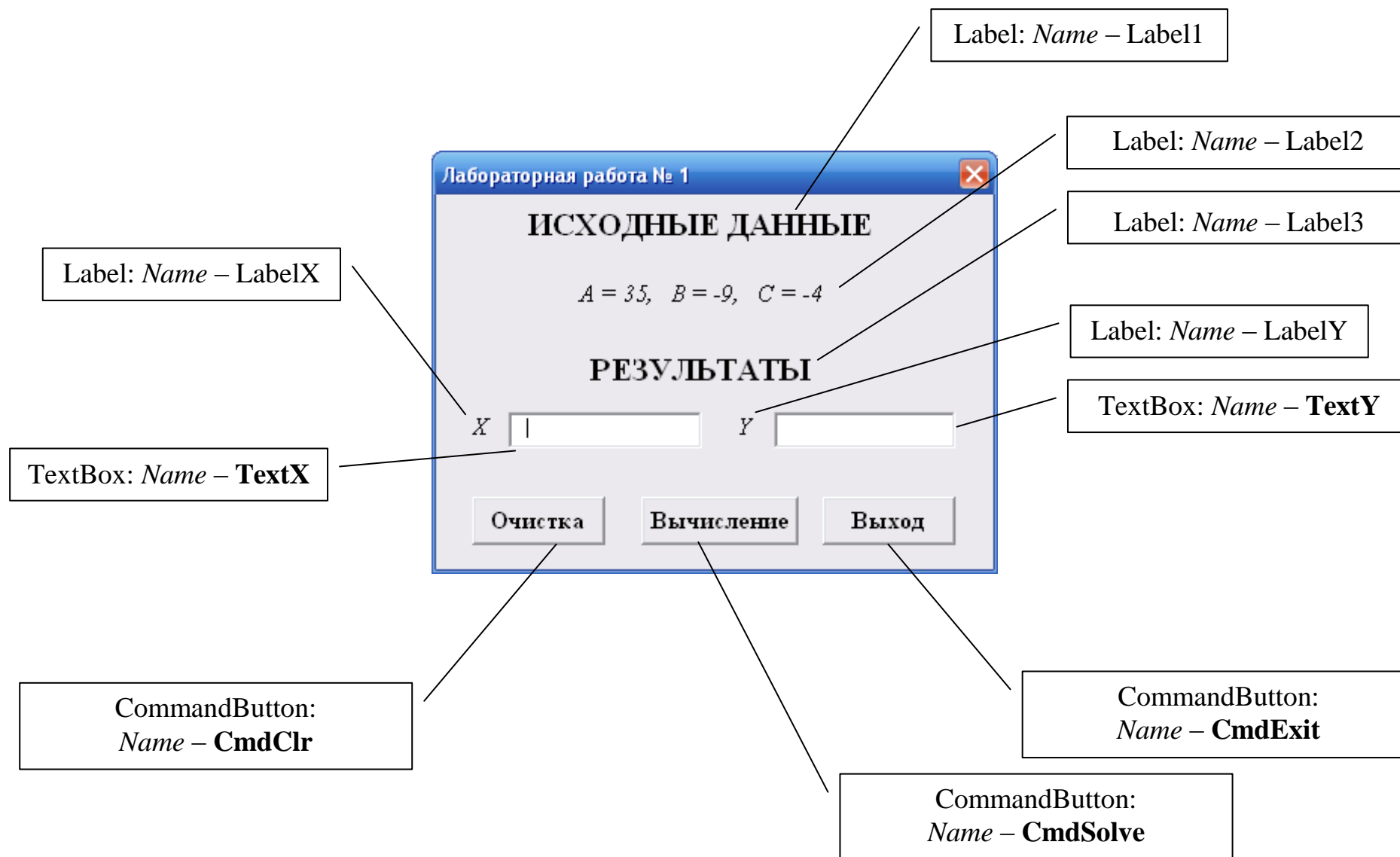
Поэтому создаем для каждой переменной свой элемент управления и меняем некоторые из его свойств:

X – элемент управления **TextBox** с такими свойствами: *Name* – **TextX**

Y – элемент управления **TextBox** с такими свойствами: *Name* – **TextY**



ЭТАП 3. Создание формы с элементами управления



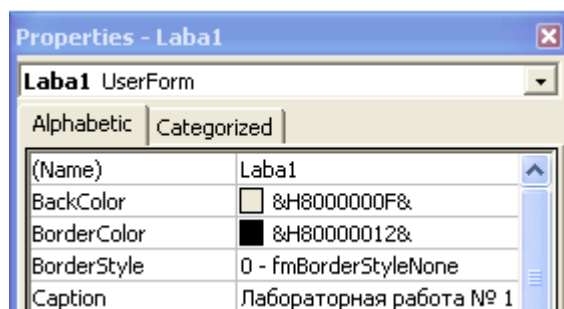
Последовательность создания формы с элементами управления

1. Запускаем программу *Microsoft Excel*
2. Запускаем встроенный редактор VBA: *Сервис* → *Макрос* → *Редактор Visual Basic*
3. Добавляем пользовательскую форму: *Insert* → *UserForm*

4. Задаем свойства формы в окне *Properties* (если данное окно отсутствует, то нужно нажать функциональную клавишу F4):

Name – ***Laba1***,

Caption – ***Лабораторная работа № 1***



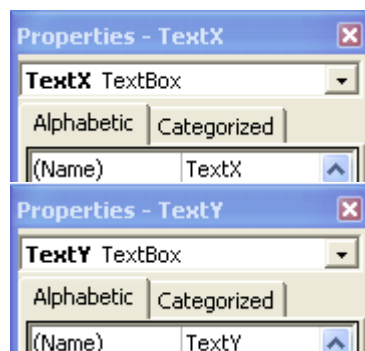
5. Добавляем элементы управления **Label** и **TextBox**, и задаем их свойства. Например,

- для переменной *X*:

Name – ***TextX***

- для переменной *Y*:

Name – ***TextY***



6. Добавляем кнопки **CommandButton** и задаем их свойства. Например,

- для кнопки *Вычисление*:

Name – ***CmdSolve***

Caption – ***Вычисление***

- для кнопки *Очистка*:

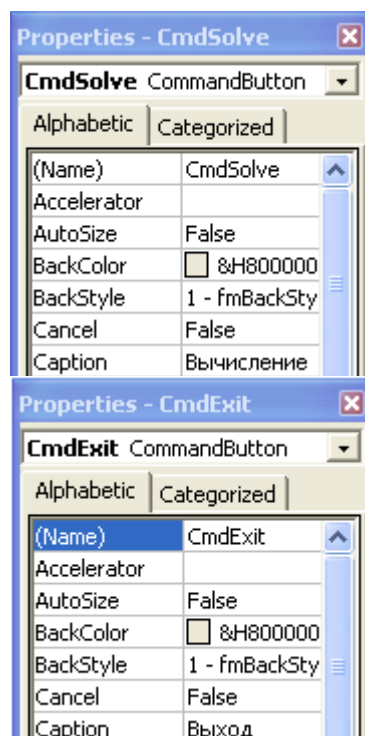
Name – ***CmdClr***

Caption – ***Очистка***

- для кнопки *Выход*:

Name – ***CmdExit***

Caption – ***Выход***



ЭТАП 4. Написание обработок событий

В созданной пользовательской форме три кнопки, следовательно, к каждой из этих кнопок необходимо написать процедуру обработки ее нажатия. Для написания обработки события «нажатия мышкой» необходимо в режиме добавления элементов управления щелкнуть дважды мышкой по соответствующей кнопке.

Очистка

Для кнопки «Очистка» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdClr_Click()  
  
End Sub
```

2. Написать код процедуры очистки всех элементов управления TextBox (в нашем случае это два элемента – **TextX** и **TextY**):

```
Private Sub CmdClr_Click()  
    TextX.Text = ""  
    TextY.Text = ""  
End Sub
```

Выход

Для кнопки «Выход» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdExit_Click()  
  
End Sub
```

2. Написать код процедуры выхода из приложения, что происходит командой End, т.е. код программы примет вид:

```
Private Sub CmdExit_Click()  
    End  
End Sub
```

Для кнопки «Вычисление» необходимо:

Вычисление

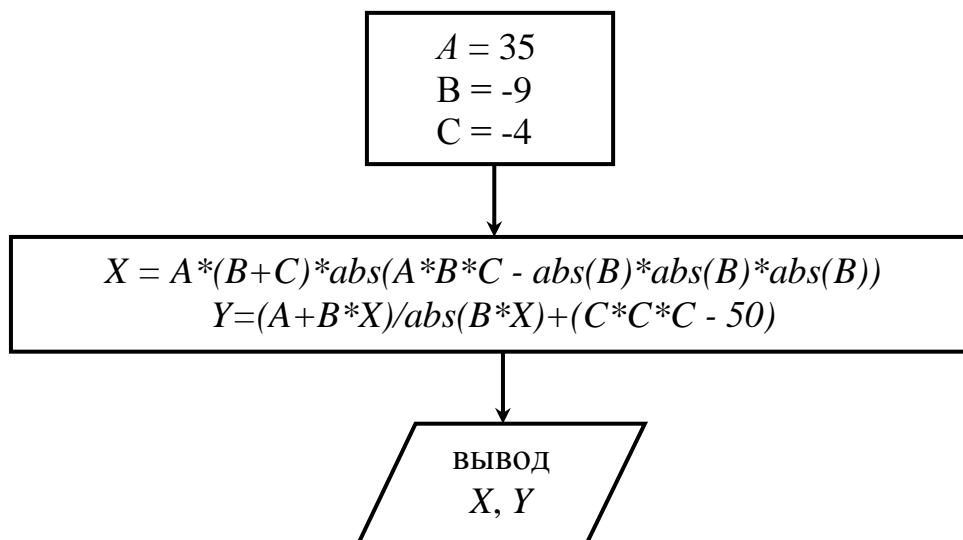
1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdSolve_Click()  
  
End Sub
```

2. Написать код процедуры согласно составленной выше блок-схеме, т.е. код программы примет вид:

```
Private Sub CmdSolve_Click()  
    A = 35  
    B = -9  
    C = -4  
    X = A*(B + C)*Abs(A*B*C - Abs(B)*Abs(B)*Abs(B))  
    Y = (A + B * X) / Abs(B * X) + (C * C * C - 50)  
    TextX.Text = CStr(X)  
    TextY.Text = CStr(Y)  
End Sub
```

Как уже отмечалось, блок-схема для данной программы выглядит так:



Поэтому фактически только последний блок вывода несколько отличается по коду и заменяется двумя операторами вывода значений переменных X , Y в соответствующие текстовые окна с именами *TextX* и *TextY*. При этом функция **CStr()** преобразует числовое значение в текстовое.

Помимо указанных выше процедур необходимо также описать все переменные участвующие в программе. Согласно блок-схеме или указанной выше программе, имеем следующие переменные:

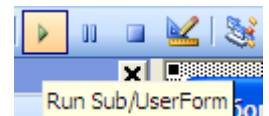
A, B, C, X, Y .

Следовательно, в программе необходимо указать следующий код описания этих переменных

```
Dim A As Single  
Dim B As Single  
Dim C As Single  
Dim X As Single  
Dim Y As Single
```

ЭТАП 5. Запуск программы, получение и проверка результатов

1. Для запуска программы необходимо нажать кнопку *Run*
2. В появившейся форме нажать кнопку *Вычисление*, при этом значения X и Y будут выведены в соответствующие текстовые поля **TextBox**.



3. Проверка правильности полученных результатов

$$X = 35 * (-9 + (-4)) \left| 35 * (-9) * (-4) - |-9|^3 \right| = -241605$$

$$Y = \frac{35 + (-9) * 241605}{|(-9) * 241605|} + ((-4)^3 - 50) = -113$$

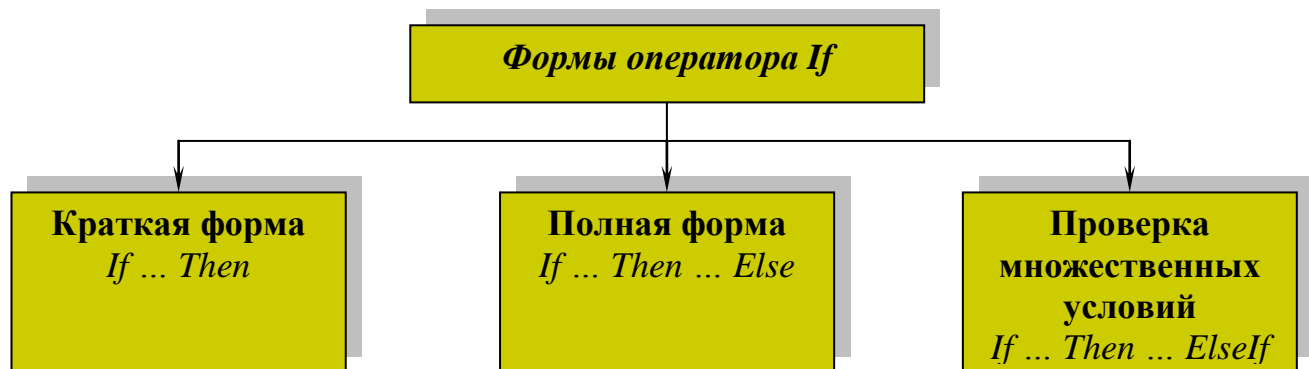
ЛАБОРАТОРНАЯ РАБОТА № 2

Тема: «Организация интерфейса для реализации разветвляющихся процессов в среде VBA»

Цель: *Получение навыков составления программ разветвляющихся вычислительных процессов.*

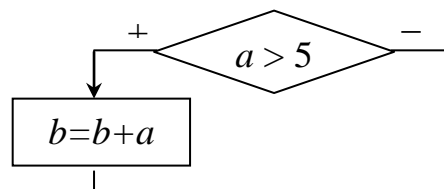
Теоретическая часть

Для реализации разветвляющегося процесса используется оператор условного перехода (*условный оператор*) *If*. Условный оператор *If* имеет три формы.



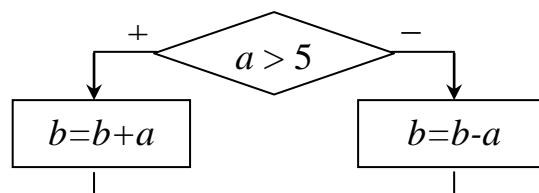
Краткая форма

If a > 5 Then b = b + a



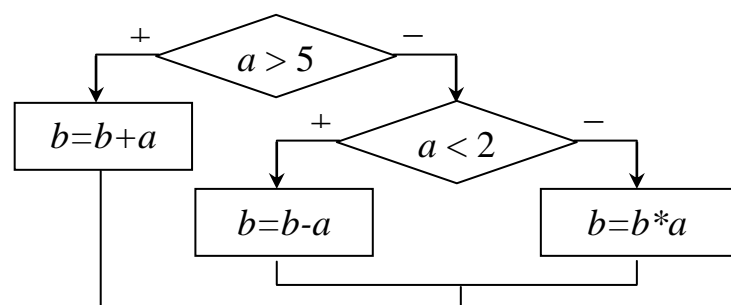
Полная форма

*If a > 5 Then
 b = b + a
Else
 b = b - a
End If*



Проверка множественных условий

*If a > 5 Then
 b = b + a
ElseIf a < 2 Then
 b = b - a
Else
 b = b * a
End If*



Пример решения задания

Задание

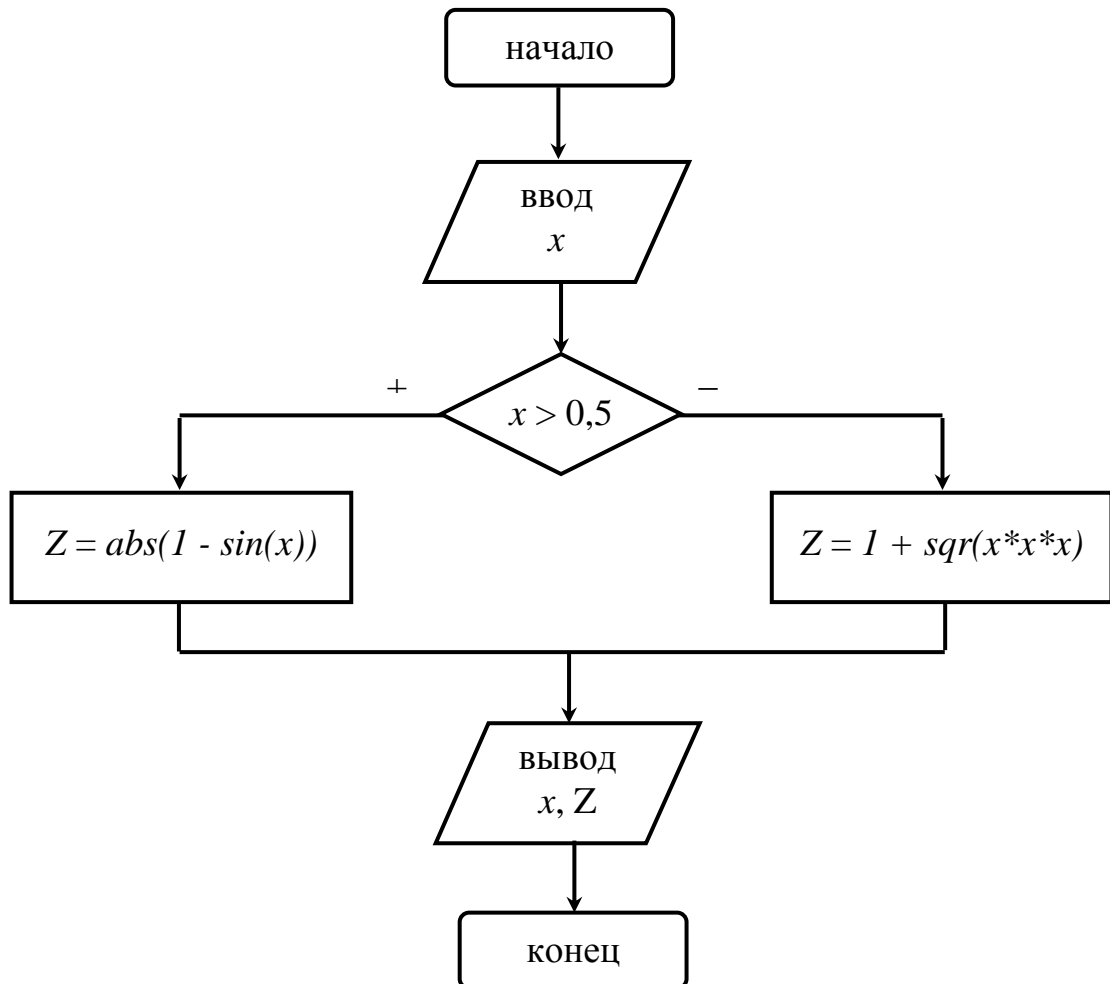
Вычислить Z , если известно условие:

$$Z = \begin{cases} |1 - \sin x|, & \text{если } x > 0,5; \\ 1 + \sqrt{x^3}, & \text{если } x \leq 0,5. \end{cases}$$

Вывести значения в виде: x, Z .

Решение поставленной задачи

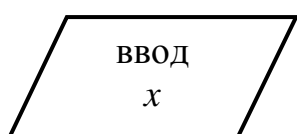
ЭТАП 1. Составление блок-схемы



ЭТАП 2. Определение элементов управления ввода-вывода

На данном этапе необходимо определиться с элементами управления, которые будут использоваться для ввода и вывода данных. Добавление элементов происходит с панели инструментов *Toolbox* (в случае отсутствия данной панели на экране необходимо: *Insert* → *Toolbox*)

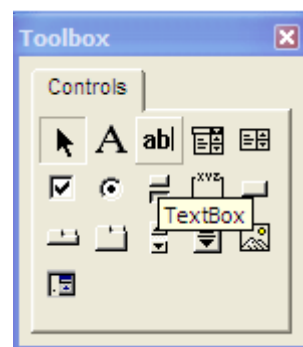
ИСХОДНЫЕ ДАННЫЕ



Наличие в блок-схеме блока ввода данных x требует добавления в форму одного

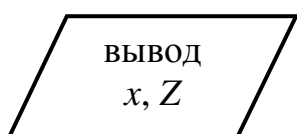
элемента управления типа **TextBox**.

Поэтому создаем для этой переменной x свой элемент управления и меняем некоторые из его свойств:



x – элемент управления TextBox с такими свойствами: *Name* – **TextX**

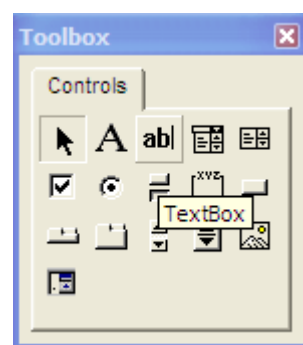
РЕЗУЛЬТАТЫ



Наличие в блок-схеме блока вывода данных x, Z требует добавления в форму двух

элементов управления типа **TextBox**.

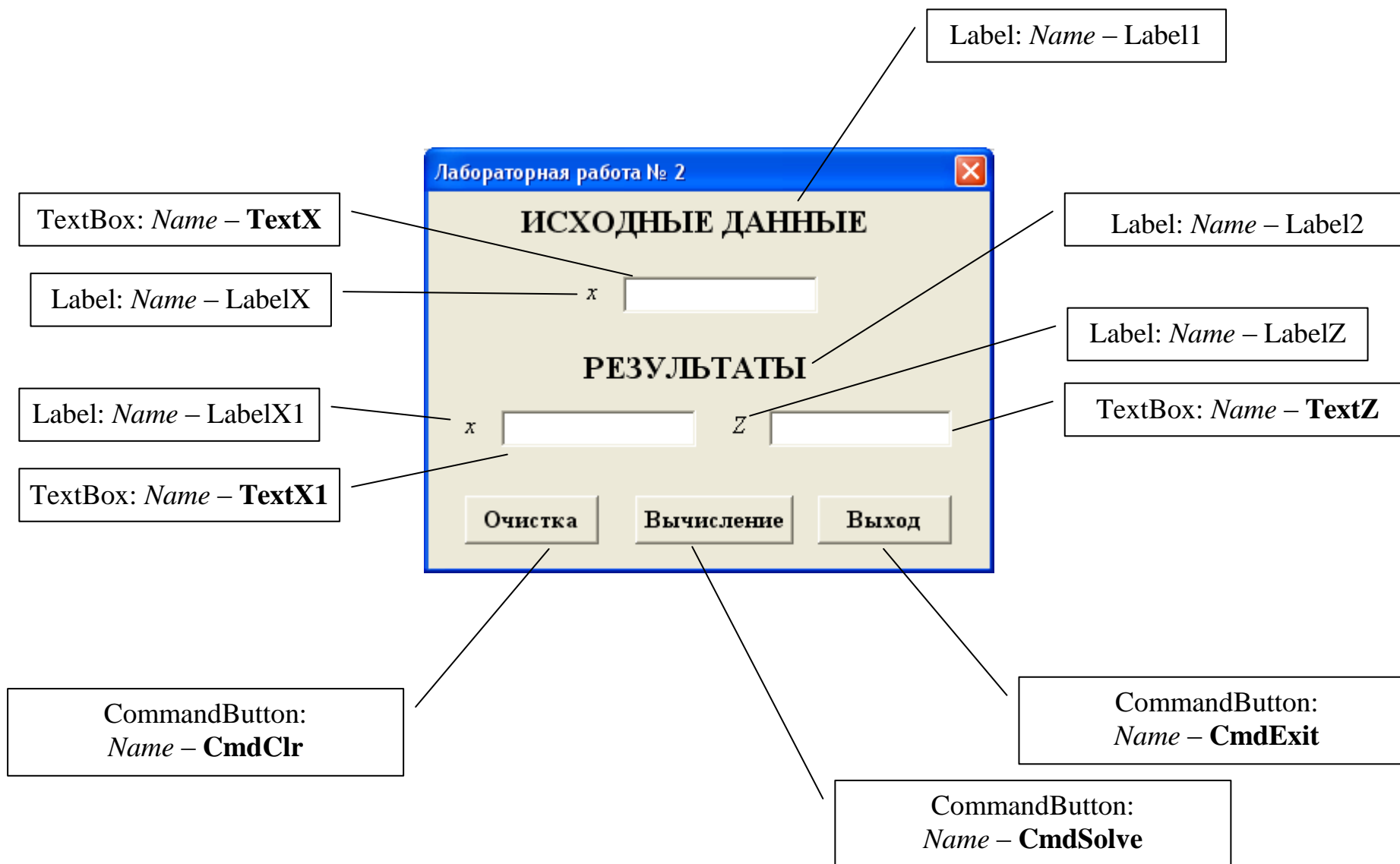
Поэтому создаем для каждой переменной свой элемент управления и меняем некоторые из его свойств:



x – элемент управления TextBox с такими свойствами: *Name* – **TextX1**

Z – элемент управления TextBox с такими свойствами: *Name* – **TextZ**

ЭТАП 3. Создание формы с элементами управления

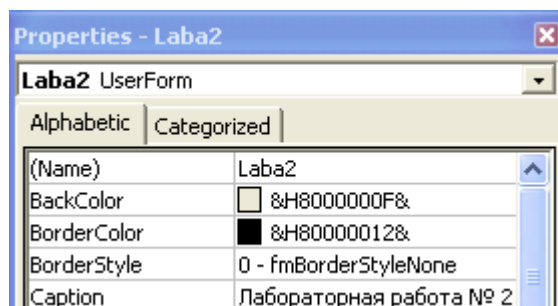


Последовательность создания формы с элементами управления

1. Запускаем программу *Microsoft Excel*
2. Запускаем встроенный редактор VBA: *Сервис* → *Макрос* → *Редактор Visual Basic*
3. Добавляем пользовательскую форму: *Insert* → *UserForm*
4. Задаем свойства формы в окне *Properties* (если данное окно отсутствует, то нужно нажать функциональную клавишу F4):

Name – **Laba2**,

Caption – **Лабораторная работа № 2**



5. Добавляем элементы управления **Label** и **TextBox**, и задаем их свойства. Например,

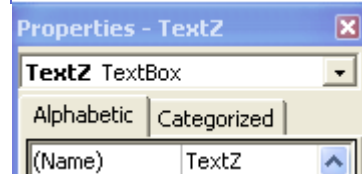
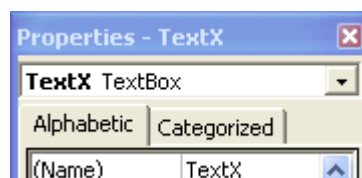
- для переменной *x* два **TextBox**:

для ввода *Name* – **TextX**

для вывода *Name* – **TextX1**

- для переменной *Z*:

Name – **TextZ**

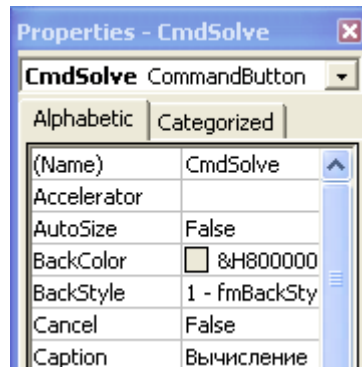


6. Добавляем кнопки **CommandButton** и задаем их свойства. Например,

- для кнопки *Вычисление*:

Name – **CmdSolve**

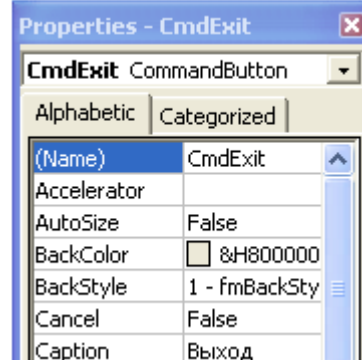
Caption – **Вычисление**



- для кнопки *Очистка*:

Name – **CmdClr**

Caption – **Очистка**



- для кнопки *Выход*:

Name – **CmdExit**

Caption – **Выход**

ЭТАП 4. Написание обработок событий

В созданной пользовательской форме три кнопки, следовательно, к каждой из этих кнопок необходимо написать процедуру обработки ее нажатия. Для написания обработки события «нажатия мышкой» необходимо в режиме добавления элементов управления щелкнуть дважды мышкой по соответствующей кнопке.

Очистка

Для кнопки «Очистка» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdClr_Click()  
  
End Sub
```

2. Написать код процедуры очистки всех элементов управления TextBox (в нашем случае это три элемента – *TextX*, *TextX1* и *TextZ*):

```
Private Sub CmdClr_Click()  
    TextX.Text = ""  
    TextX1.Text = ""  
    TextZ.Text = ""  
End Sub
```

Выход

Для кнопки «Выход» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdExit_Click()  
  
End Sub
```

2. Написать код процедуры выхода из приложения, что происходит командой End, т.е. код программы примет вид:

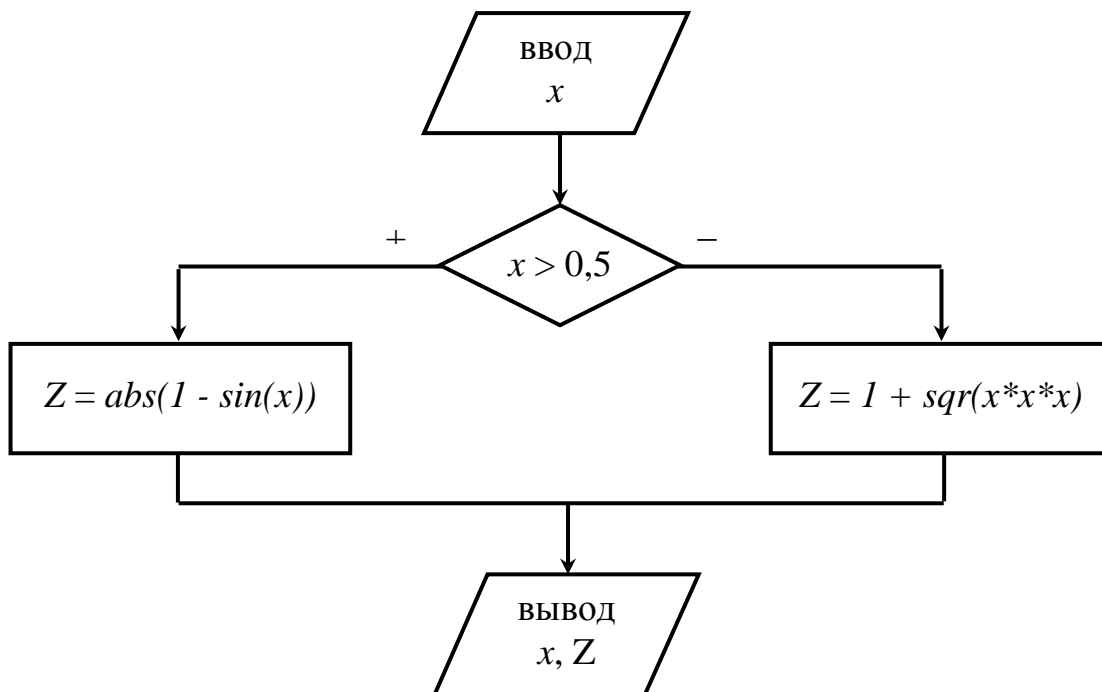
```
Private Sub CmdExit_Click()  
    End  
End Sub
```

Для кнопки «Вычисление» необходимо:

1. Произвести двойной щелчок по кнопке и необходимо ввести код процедуры, согласно блок-схеме, т.е. код программы примет вид:

```
Private Sub CmdSolve_Click()  
    X = CSng(TextX.Text)  
    If X > 0.5 Then  
        Z = Abs(1 - Sin(X))  
    Else  
        Z = 1 - Sqr(X * X * X)  
    End If  
    TextX1.Text = CStr(X)  
    TextZ.Text = CStr(Z)  
End Sub
```

Как уже отмечалось, блок-схема для данной программы выглядит так:



Для ввода значения переменной x используется функция **CSng()**, которая преобразует текстовое значение в числовое. Разветвляющийся процесс реализуется с помощью оператора **If**. Вывод значений переменных x , Z в соответствующие текстовые окна с именами *TextX1* и *TextZ* происходит с помощью функции **CStr()**, которая преобразует числовое значение в текстовое.

Помимо указанных выше процедур необходимо также описать все переменные участвующие в программе. Согласно блок-схеме или указанной выше программе, имеем следующие переменные:

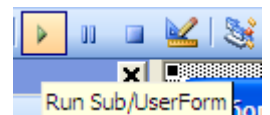
x , Z .

Следовательно, в программе необходимо указать следующий код описания этих переменных

```
Dim X As Single  
Dim Z As Single
```

ЭТАП 5. Запуск программы, получение и проверка результатов

1. Для запуска программы необходимо нажать кнопку *Run*
2. В появившейся форме заполнить ИСХОДНЫЕ ДАННЫЕ: x



3. В появившейся форме нажать кнопку *Вычисление*, при этом значения x и Z будут выведены в соответствующие текстовые поля **TextBox**.

A screenshot of a Windows application window titled 'Лабораторная работа № 2'. The window has a blue title bar and a close button. The main area is divided into two sections: 'ИСХОДНЫЕ ДАННЫЕ' (Initial Data) and 'РЕЗУЛЬТАТЫ' (Results). Under 'ИСХОДНЫЕ ДАННЫЕ', there is a label 'x' followed by a text box containing the value '0'. Under 'РЕЗУЛЬТАТЫ', there are two text boxes: one labeled 'x' containing '0' and another labeled 'Z' containing '1'. The 'Z' text box is circled in red. At the bottom of the window, there are three buttons: 'Очистка' (Clear), 'Вычисление' (Calculation), and 'Выход' (Exit). A red arrow points from the 'Z' text box to the text below.

4. Проверка правильности полученных результатов:

$x = 0$, тогда Z будет вычисляться по второй ветви (т.к. удовлетворяет условию $x \leq 0,5$)

Имеем $Z = 1 + 0 = 1$

ЛАБОРАТОРНАЯ РАБОТА № 3

Тема: «Организация интерфейса для реализации циклических процессов в среде VBA. Цикл с предусловием»

Цель: *Получение навыков составления программ циклических вычислительных процессов. Изучение цикла с предусловием.*

Теоретическая часть

Для реализации циклического процесса с предусловием используется оператор ***While...Wend***. Данный оператор позволяет сначала проверить условие, а потом выполнять операторы цикла, поэтому если условие будет не верно, то операторы цикла выполняться не будут.

Синтаксис оператора:

While <условие>
 <оператор 1>
 <оператор 2>
 ...
 <оператор n>
Wend

Пример решения задания

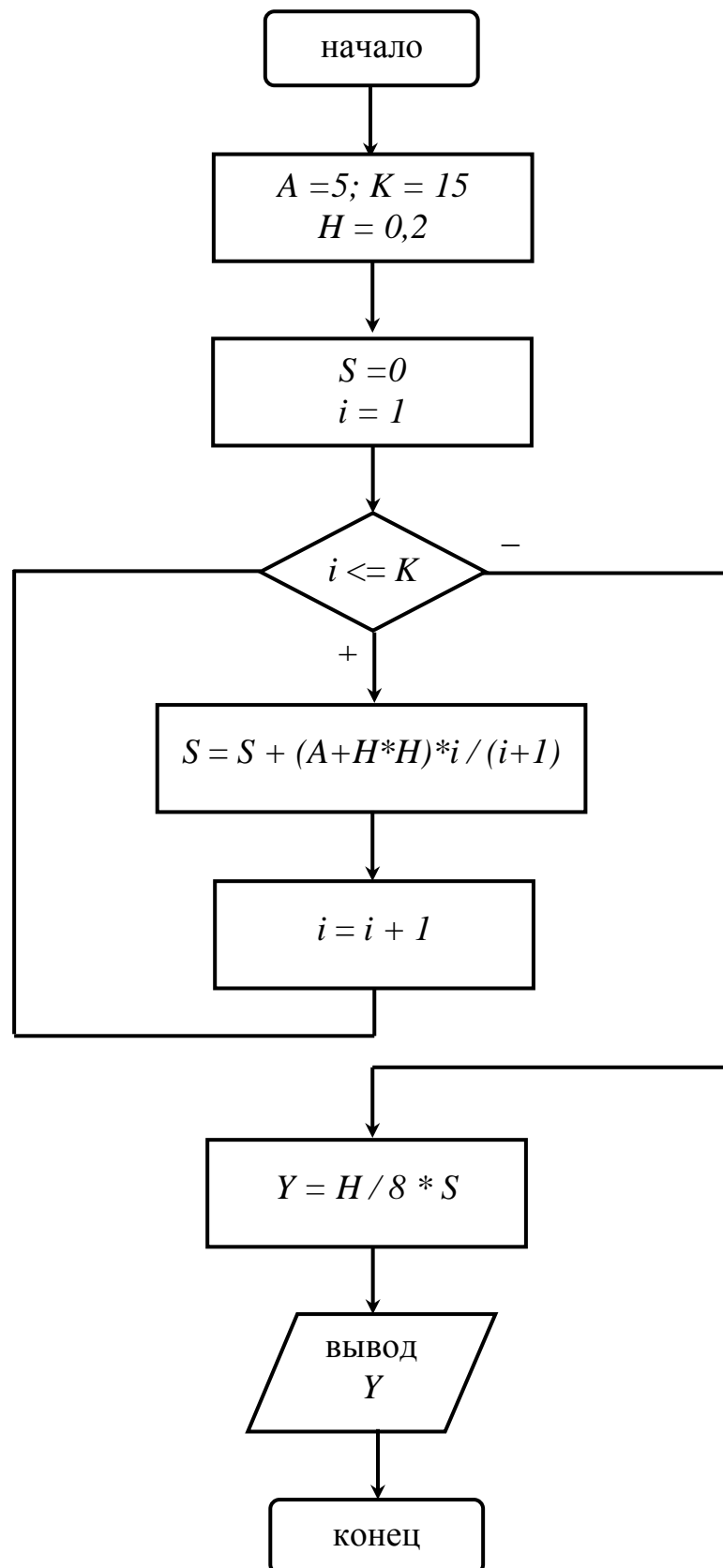
Задание

Вычислить

$$Y = \frac{H}{8} \sum_{i=1}^K \frac{(A + H^2)}{i + 1}, \text{ где } A=5, K=15, H=0,2.$$

Решение поставленной задачи

ЭТАП 1. Составление блок-схемы

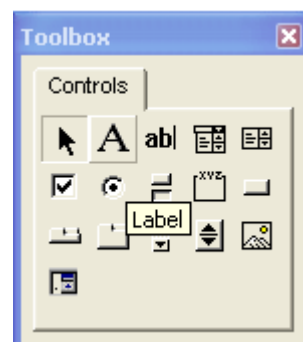


ЭТАП 2. Определение элементов управления ввода-вывода

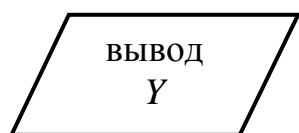
На данном этапе необходимо определиться с элементами управления, которые будут использоваться для ввода и вывода данных. Добавление элементов происходит с панели инструментов *Toolbox* (в случае отсутствия данной панели на экране необходимо: *Insert* → *Toolbox*)

ИСХОДНЫЕ ДАННЫЕ

В данном случае нет неизвестных данных, требующих ввода, поэтому для указания значений известных переменных A , K , H можно использовать элемент управления **Label**. Для данного элемента управления свойства можно не менять, т. к. он не будет использоваться в программе.

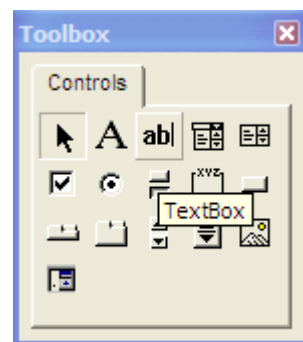


РЕЗУЛЬТАТЫ



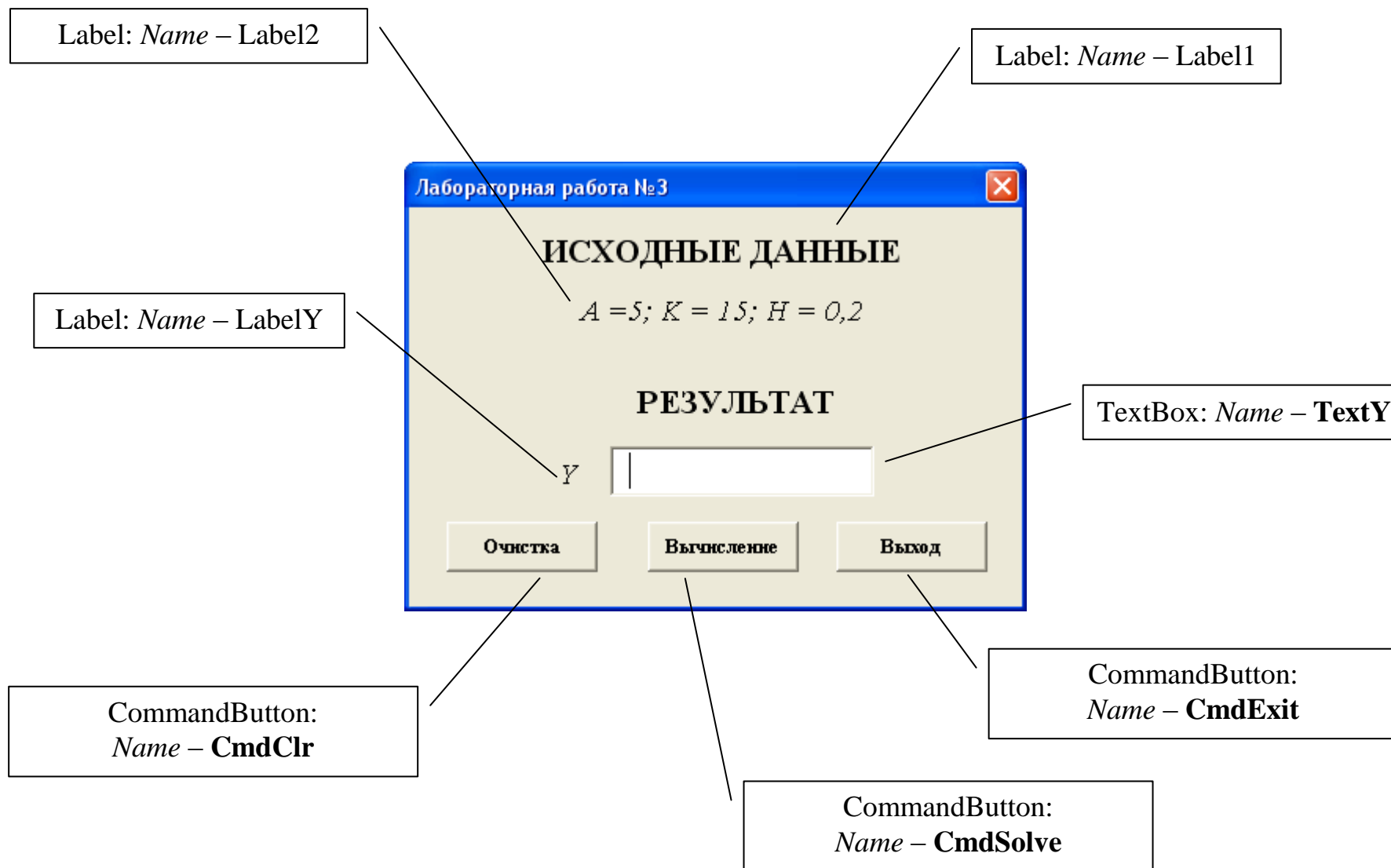
Наличие в блок-схеме блока вывода данных Y требует добавления в форму одного элемента управления типа **TextBox**.

Поэтому создаем для переменной Y свой элемент управления и меняем некоторые из его свойств:



Y – элемент управления **TextBox** с такими свойствами: *Name* – **TextY**

ЭТАП 3. Создание формы с элементами управления

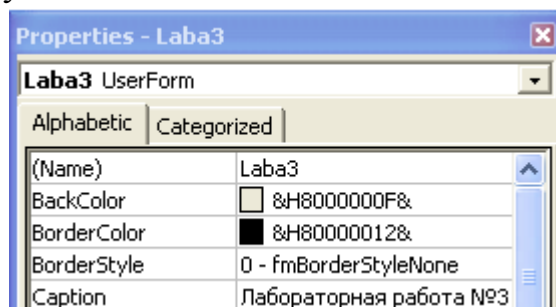


Последовательность создания формы с элементами управления

1. Запускаем программу *Microsoft Excel*
2. Запускаем встроенный редактор VBA: *Сервис* → *Макрос* → *Редактор Visual Basic*
3. Добавляем пользовательскую форму: *Insert* → *UserForm*
4. Задаем свойства формы в окне *Properties* (если данное окно отсутствует, то нужно нажать функциональную клавишу F4):

Name – **Laba3**,

Caption – **Лабораторная работа № 3**



5. Добавляем элементы управления **Label** и **TextBox**, и задаем их свойства. Например,

– для переменной *Y*:

Name – **TextY**

6. Добавляем кнопки **CommandButton** и задаем их свойства. Например,

– для кнопки *Вычисление*:

Name – **CmdSolve**

Caption – **Вычисление**

– для кнопки *Очистка*:

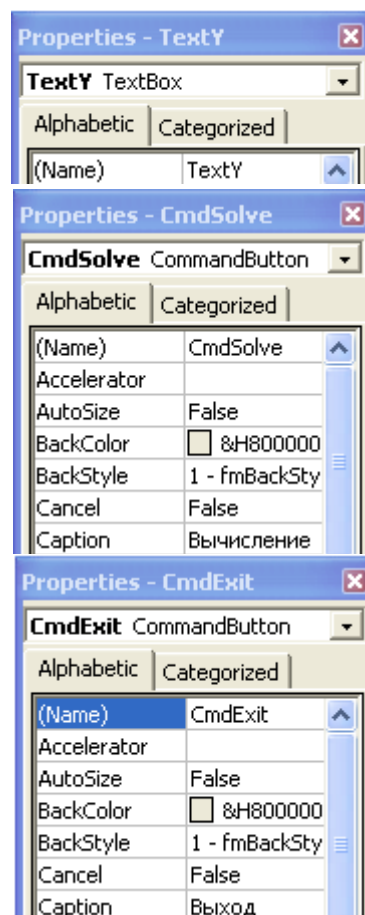
Name – **CmdClr**

Caption – **Очистка**

– для кнопки *Выход*:

Name – **CmdExit**

Caption – **Выход**



ЭТАП 4. Написание обработок событий

В созданной пользовательской форме три кнопки, следовательно. К каждой из этих кнопок необходимо написать процедуру обработки ее нажатия. Для написания обработки события «нажатия мышкой» необходимо в режиме добавления элементов управления щелкнуть дважды мышкой по соответствующей кнопке.

Очистка

Для кнопки «Очистка» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdClr_Click()  
  
End Sub
```

2. Написать код процедуры очистки всех элементов управления TextBox (в нашем случае это один элемент – **TextY**):

```
Private Sub CmdClr_Click()  
    TextY.Text = ""  
End Sub
```

Выход

Для кнопки «Выход» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdExit_Click()  
  
End Sub
```

2. Написать код процедуры выхода из приложения, что происходит командой End, т.е. код программы примет вид:

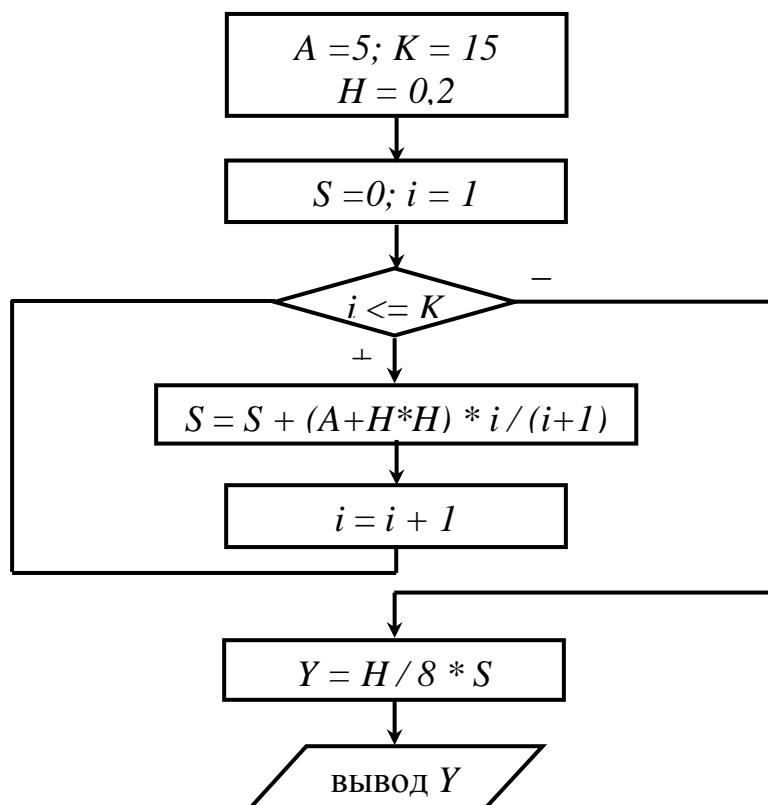
```
Private Sub CmdExit_Click()  
    End  
End Sub
```

Для кнопки «Вычисление» необходимо:

1. Произвести двойной щелчок по кнопке и необходимо ввести код процедуры, согласно блок-схеме, т.е. код программы примет вид:

```
Private Sub CmdSolve_Click()
    A = 5
    K = 15
    H = 0.2
    S = 0
    i = 1
    While i <= K
        S = S + (A + H * H) * i / (i + 1)
        i = i + 1
    Wend
    Y = H / 8 * S
    TextY.Text = CStr(Y)
End Sub
```

Как уже отмечалось, блок-схема для данной программы выглядит так:



Циклический процесс реализуется с помощью оператора *While...Wend*. Вывод значения переменной *Y* происходит в текстовое окно с именем *TextY* с помощью функции **CStr()**, которая преобразует числовое значение в текстовое.

Помимо указанных выше процедур необходимо также описать все переменные участвующие в программе. Согласно блок-схеме или указанной выше программе, имеем следующие переменные:

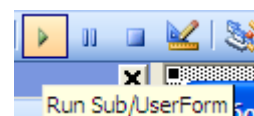
Y, A, K, H, S, i

Следовательно, в программе необходимо указать следующий код описания этих переменных

```
Dim i As Single  
Dim S As Single  
Dim Y As Single  
Dim A As Single  
Dim K As Single  
Dim H As Single
```

ЭТАП 5. Запуск программы, получение и проверка результатов

1. Для запуска программы необходимо нажать кнопку *Run*
2. В появившейся форме нажать кнопку *Вычисление*, при этом значение Y будет выведено в текстовое поле **TextBox**.



3. Проверка правильности полученных результатов:

$$Y = \frac{0,2}{8} \sum_{i=1}^K \frac{(5 + 0,2^2)i}{i+1} = 0,025 * \left(\frac{5,04}{2} + \frac{5,04 \cdot 2}{3} + \frac{5,04 \cdot 3}{4} + \dots + \frac{5,04 \cdot 15}{16} \right) = 1,590028$$

ЛАБОРАТОРНАЯ РАБОТА № 4

Тема: «Организация интерфейса для реализации циклических процессов в среде VBA. Цикл с постусловием»

Цель: *Получение навыков составления программ циклических вычислительных процессов. Изучение цикла с постусловием*

Теоретическая часть

Для реализации циклического процесса с постусловием используется оператор ***Do...Loop Until***. Данный оператор сначала выполняет операторы цикла, а потом проверяет условие выхода из цикла, если оно выполняется, то происходит выход из цикла. Таким образом, данный цикл всегда выполняется хотя бы один раз. Синтаксис оператора:

Do

<оператор 1>

<оператор 2>

...

<оператор n>

Loop Until <условие>

Для вывода значений из программы на лист Microsoft Excel необходимо использовать функцию Cells. Синтаксис функции

Cells(номер строки, номер столбца)

Например, Cells(3,4)=x выведет значение переменной x в ячейку D3

Пример решения задания

Задание

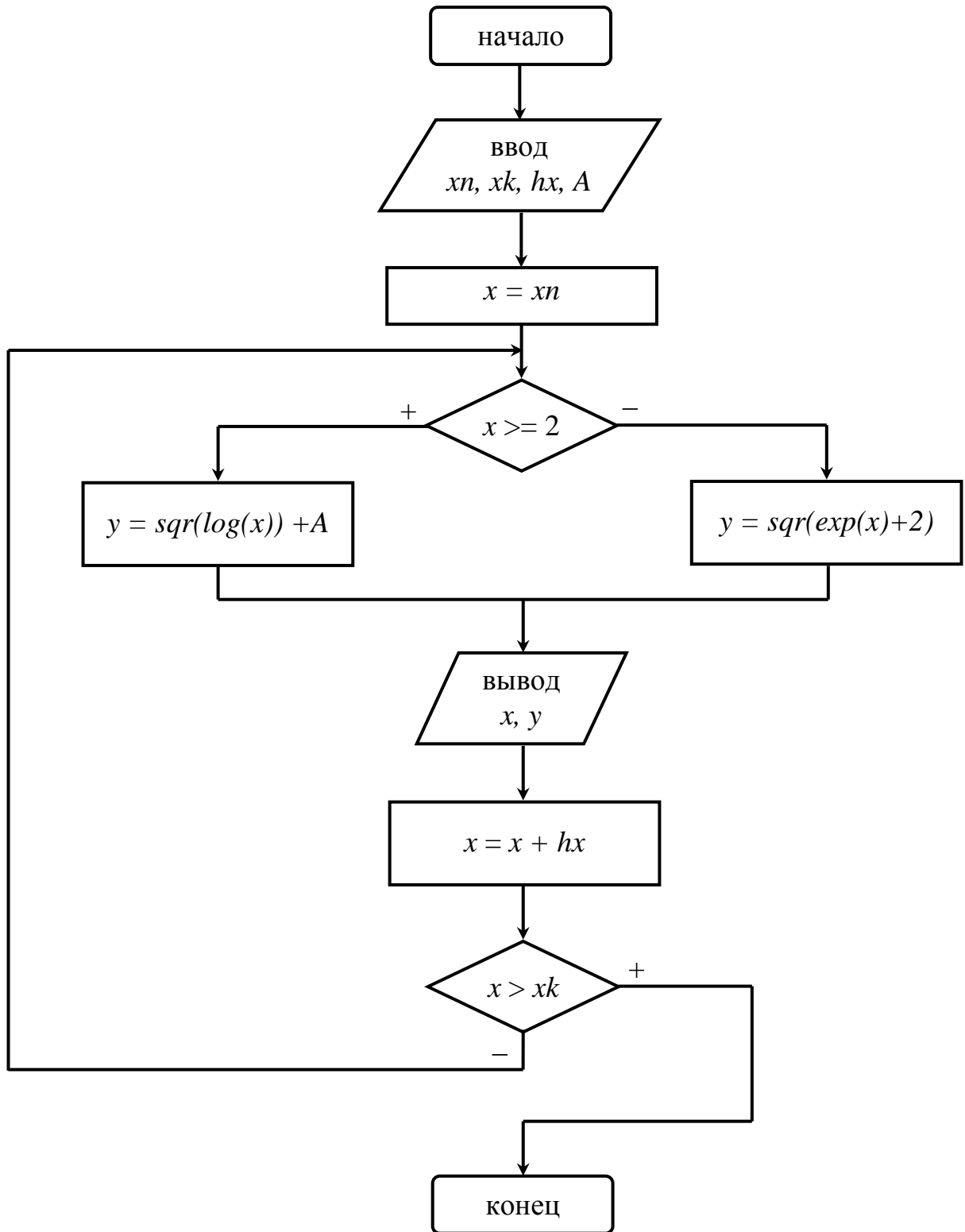
Вычислите Y, если известно условие:

$$Y = \begin{cases} \sqrt{\ln x} + A, & \text{если } x \geq 2; \\ \sqrt{e^x + 2}, & \text{если } x < 2. \end{cases} \quad x \in [-4; 4]; \Delta x = 0,5; A = 2.$$

Вывести значения в виде: x, Y.

Решение поставленной задачи

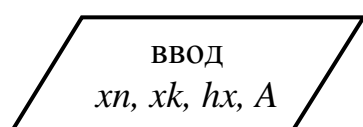
ЭТАП 1. Составление блок-схемы



ЭТАП 2. Определение элементов управления ввода-вывода

На данном этапе необходимо определиться с элементами управления, которые будут использоваться для ввода и вывода данных. Добавление элементов происходит с панели инструментов *Toolbox* (в случае отсутствия данной панели на экране необходимо: *Insert* → *Toolbox*)

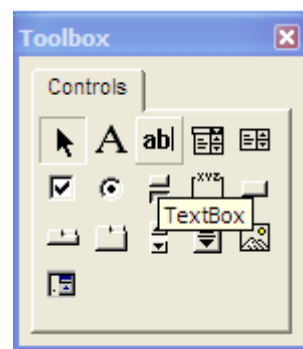
ИСХОДНЫЕ ДАННЫЕ



Наличие в блок-схеме блока ввода данных x_n, x_k, h_x, A требует добавления в форму четырех

элементов управления типа **TextBox**.

Поэтому создаем для переменных x_n, x_k, h_x, A соответствующие элементы управления и меняем некоторые из его свойств:



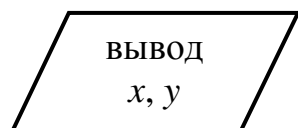
x_n – элемент управления TextBox с такими свойствами: *Name* – **TextXn**

x_k – элемент управления TextBox с такими свойствами: *Name* – **TextXk**

h_x – элемент управления TextBox с такими свойствами: *Name* – **TextHx**

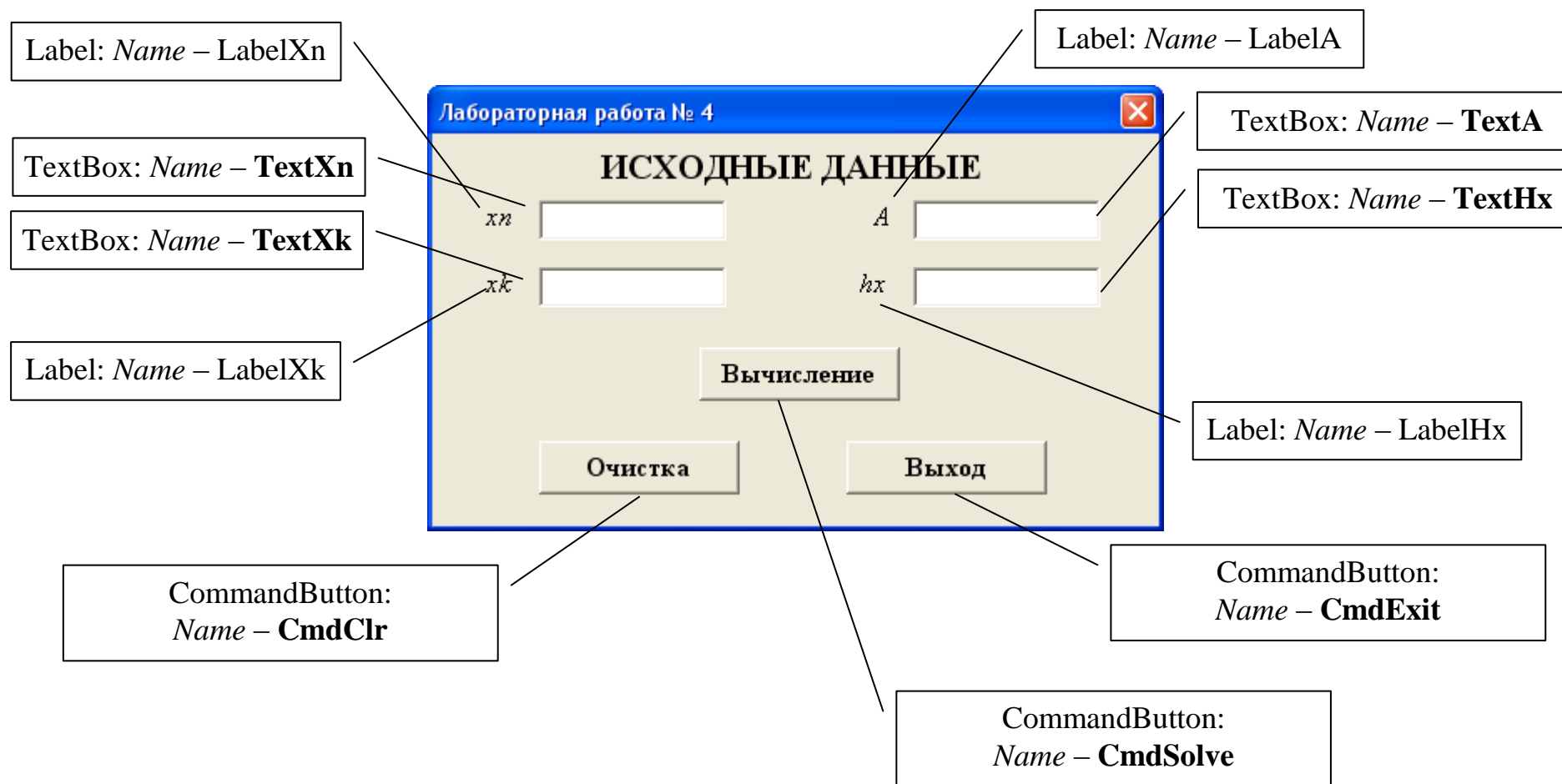
A – элемент управления TextBox с такими свойствами: *Name* – **TextA**

РЕЗУЛЬТАТЫ



В результате выполнения программы будет получен набор значений x, y . Т.к. эти значения не единичны, то их вывод будет осуществлен в ячейки листа Microsoft Excel, поэтому для вывода результатов в форме не потребуется элементов управления.

ЭТАП 3. Создание формы с элементами управления

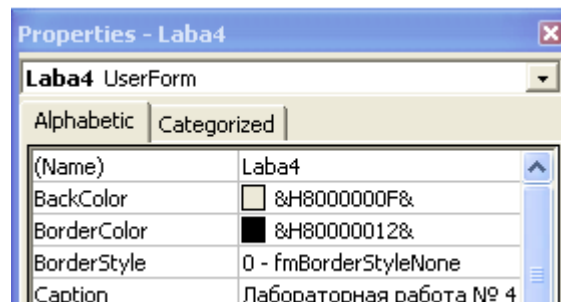


Последовательность создания формы с элементами управления

1. Запускаем программу *Microsoft Excel*
2. Запускаем встроенный редактор VBA: *Сервис* → *Макрос* → *Редактор Visual Basic*
3. Добавляем пользовательскую форму: *Insert* → *UserForm*
4. Задаем свойства формы в окне *Properties* (если данное окно отсутствует, то нужно нажать функциональную клавишу F4):

Name – **Laba4**,

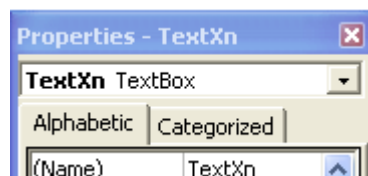
Caption – **Лабораторная работа № 4**



5. Добавляем элементы управления **Label** и **TextBox**, и задаем их свойства. Например,

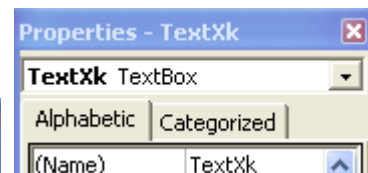
- для переменной *xn*:

для ввода *Name* – **TextXn**



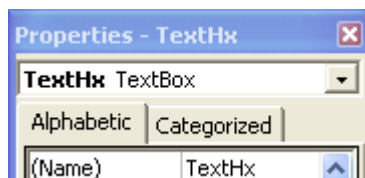
- для переменной *xk*:

Name – **TextXk**



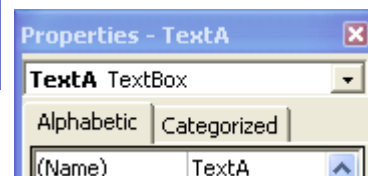
- для переменной *hx*:

Name – **TextHx**



- для переменной *A*:

Name – **TextA**



6. Добавляем кнопки **CommandButton** и задаем их свойства. Например,

- для кнопки *Вычисление*:

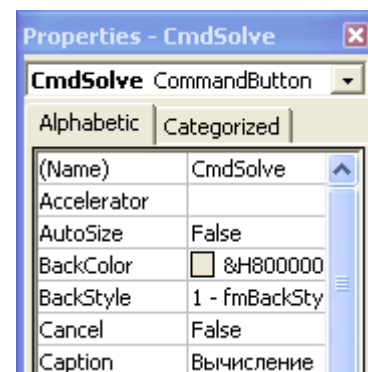
Name – **CmdSolve**, *Caption* – **Вычисление**

- для кнопки *Очистка*:

Name – **CmdClr**, *Caption* – **Очистка**

- для кнопки *Выход*:

Name – **CmdExit**, *Caption* – **Выход**



ЭТАП 4. Написание обработок событий

В созданной пользовательской форме три кнопки, следовательно, к каждой из этих кнопок необходимо написать процедуру обработки ее нажатия. Для написания обработки события «нажатия мышкой» необходимо в режиме добавления элементов управления щелкнуть дважды мышкой по соответствующей кнопке.

Для кнопки «Очистка» необходимо:

Очистка

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdClr_Click()  
  
End Sub
```

2. Написать код процедуры очистки всех элементов управления TextBox (в нашем случае это четыре элемента – *TextXn*, *TextXk*, *TextHx* и *TextA*):

```
Private Sub CmdClr_Click()  
    TextXn.Text = "": TextXk.Text = ""  
    TextHx.Text = "": TextA.Text = ""  
End Sub
```

Для кнопки «Выход» необходимо:

Выход

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdExit_Click()  
  
End Sub
```

2. Написать код процедуры выхода из приложения, что происходит командой End, т.е. код программы примет вид:

```
Private Sub CmdExit_Click()  
    End  
End Sub
```

Для кнопки «Вычисление» необходимо:

Вычисление

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdSolve_Click()  
  
End Sub
```

2. Написать код процедуры согласно составленной выше блок-схеме, т.е. код программы примет вид:

```
Private Sub CmdSolve_Click()  
    xn = CSng(TextXn.Text)  
    xk = CSng(TextXk.Text)  
    hx = CSng(TextHx.Text)  
    A = CSng(TextA.Text)  
    Cells(1, 1) = "x"  
    Cells(1, 2) = "y"  
    i = 2  
    x = xn  
    Do  
        If x > 2 Then  
            Y = Sqr(Log(x)) + A  
        Else  
            Y = Sqr(Exp(x) + 2)  
        End If  
        Cells(i, 1) = x: Cells(i, 2) = Y  
        i = i + 1  
        x = x + hx  
    Loop Until x > xk  
End Sub
```

Для ввода значений переменных x_n , x_k , h_x , A используется функция **CSng()**, которая преобразует текстовое значение в числовое. Циклический процесс реализуется с помощью оператора **Do...Loop Until**. Как уже отмечалось вывод значений переменных x , y происходит на лист Microsoft Excel, поэтому для вывода используется функция **Cells(номер строки, номер столбца)**. В примере для вывода на лист Microsoft Excel в код программы добавлены строки выделенные рамкой.

Помимо указанных выше процедур необходимо также описать все переменные участвующие в программе. Согласно блок-схеме или указанной выше программе, имеем следующие переменные:

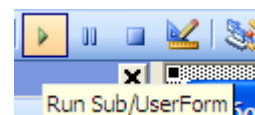
xn, xk, hx, A, x, Y, i .

Следовательно, в программе необходимо указать следующий код описания этих переменных

```
Dim A As Single
Dim xn As Single, xk As Single
Dim hx As Single, x As Single, Y As Single
Dim i As Integer
```

ЭТАП 5. Запуск программы, получение и проверка результатов

1. Для запуска программы необходимо нажать кнопку *Run*
2. В появившейся форме заполнить ИСХОДНЫЕ ДАННЫЕ: xn, xk, hx, A



3. В появившейся форме нажать кнопку *Вычисление*, при этом значения x и Y будут выведены на Лист Excel.

	A	B
1	x	Y
2	-4	1,420674
3	-3,5	1,42485
4	-3	1,431708
5	-2,5	1,442943
6	-2	1,461279
7	-1,5	1,491016
8	-1	1,538792
9	-0,5	1,614475
10	0	1,732051
11	0,5	1,910163
12	1	2,172161
13	1,5	2,545916
14	2	3,064157
15	2,5	2,957231
16	3	3,048147
17	3,5	3,119269
18	4	3,17741
19		

4. Проверка правильности полученных "некоторых" результатов:

$x = 0$, тогда Y будет вычисляться по второй ветви (т.к. удовлетворяет условию $x < 2$)

Имеем

$$Y = \sqrt{e^0 + 2} = \sqrt{1 + 2} = \sqrt{3} \quad Z = 1,732051$$

Значение x от -4 до 4

ЛАБОРАТОРНАЯ РАБОТА № 5

Тема: «Организация интерфейса для реализации циклических процессов в среде VBA. Цикл с блоком итераций»

Цель: *Получение навыков составления программ циклических вычислительных процессов. Изучение цикла с известным количеством повторений (блоком итераций)*

Теоретическая часть

Для реализации циклического процесса с известным количеством повторений (блок итераций) постусловием используется оператор **For...Next**. Данный оператор сначала выполняет операторы цикла, а потом проверяет условие выхода из цикла, если оно выполняется, то происходит выход из цикла. Таким образом, данный цикл всегда выполняется хотя бы один раз. Синтаксис оператора:

For *Счетчик=НачЗначение To КонЗначение [Step Приращение]*

<оператор 1>

<оператор 2>

...

<оператор n>

Next [*Счетчик*]

Пример решения задания

Задание

Вычислить W

$$W = \frac{\sum_{k=1}^6 g_k}{y} + \frac{\prod_{j=2}^5 (a_j + z^2)}{z}, \quad g_k, a_j - \text{известные, } y = 5; z = 3.$$

Ход решения задания

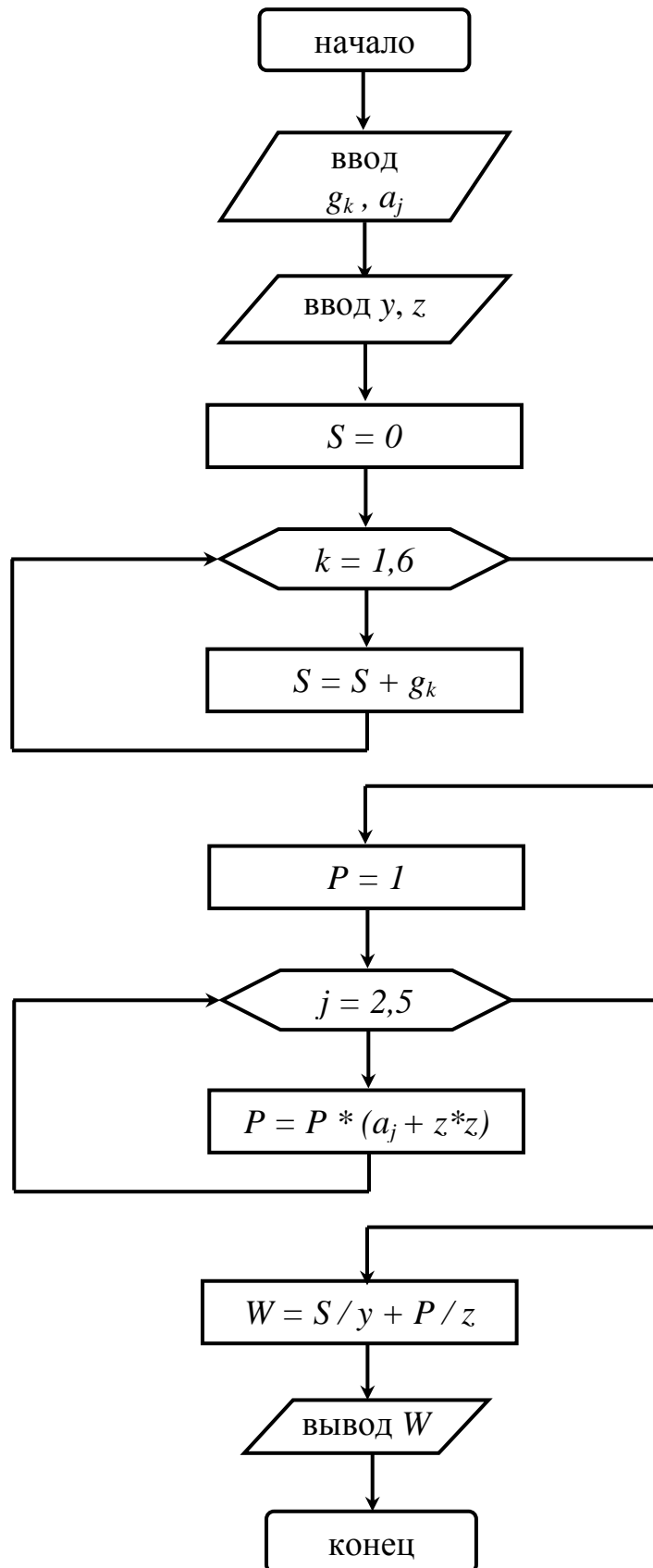
Вводим обозначения:

$$S = \sum_{k=1}^6 g_k \quad \text{и} \quad P = \prod_{j=2}^5 (a_j + z^2), \quad \text{тогда} \quad W = \frac{S}{y} + \frac{P}{z}.$$

Для расчета каждой из полученных новых переменных S и P возможно использование циклической схемы с блоком итераций.

Решение поставленной задачи

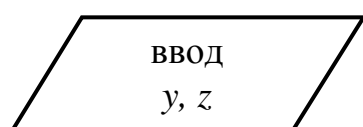
ЭТАП 1. Составление блок-схемы



ЭТАП 2. Определение элементов управления ввода-вывода

На данном этапе необходимо определиться с элементами управления, которые будут использоваться для ввода и вывода данных. Добавление элементов происходит с панели инструментов *Toolbox* (в случае отсутствия данной панели на экране необходимо: *Insert* → *Toolbox*)

ИСХОДНЫЕ ДАННЫЕ



Наличие в блок-схеме блока ввода данных y, z требует добавления в форму двух элементов

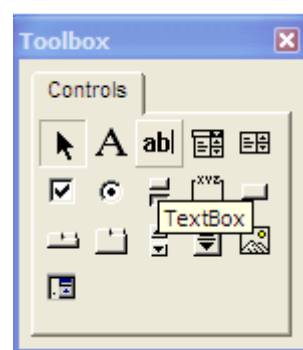
управления типа **TextBox**.

Поэтому создаем для переменных y, z соответствующие элементы управления и меняем некоторые из его свойств:

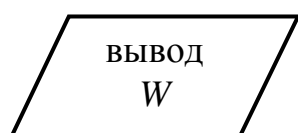
y – элемент управления TextBox с такими свойствами: *Name* – **TextY**

z – элемент управления TextBox с такими свойствами: *Name* – **TextZ**

Ввод массивов g_k, a_j происходит с листа Microsoft Excel, поэтому для их ввода не требуется элементов управления.



РЕЗУЛЬТАТЫ

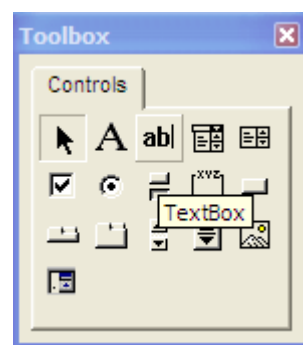


Наличие в блок-схеме блока вывода данных W требует добавления в форму одного

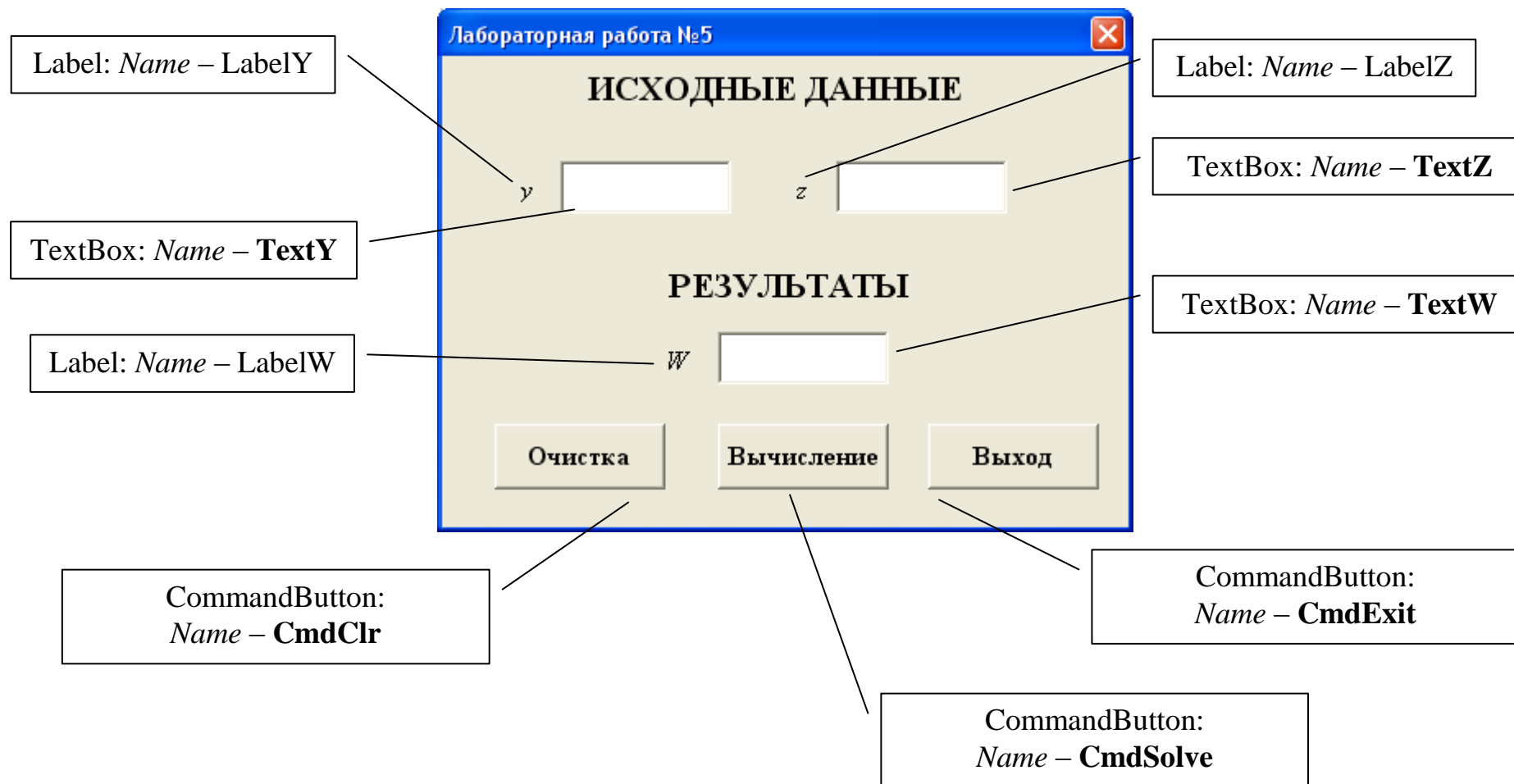
элемента управления типа **TextBox**.

Поэтому создаем для переменной W свой элемент управления и меняем некоторые из его свойств:

W – элемент управления TextBox с такими свойствами: *Name* – **TextW**



ЭТАП 3. Создание формы с элементами управления

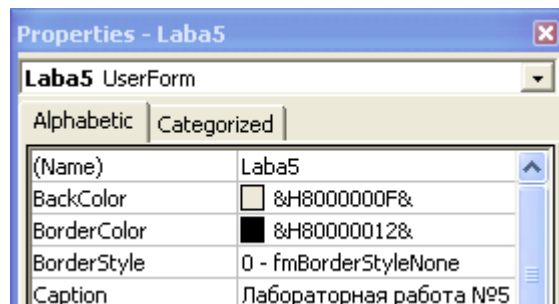


Последовательность создания формы с элементами управления

1. Запускаем программу *Microsoft Excel*
2. Запускаем встроенный редактор VBA: *Сервис* → *Макрос* → *Редактор Visual Basic*
3. Добавляем пользовательскую форму: *Insert* → *UserForm*
4. Задаем свойства формы в окне *Properties* (если данное окно отсутствует, то нужно нажать функциональную клавишу F4):

Name – **Laba5**,

Caption – **Лабораторная работа № 5**



5. Добавляем элементы управления **Label** и **TextBox**, и задаем их свойства. Например,

– для переменной *y*:

Name – **TextY**

– для переменной *z*:

Name – **TextZ**

– для переменной *W*:

Name – **TextW**

6. Добавляем кнопки **CommandButton** и задаем их свойства. Например,

– для кнопки *Вычисление*:

Name – **CmdSolve**

Caption – **Вычисление**

– для кнопки *Очистка*:

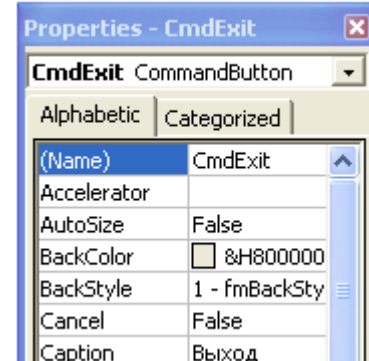
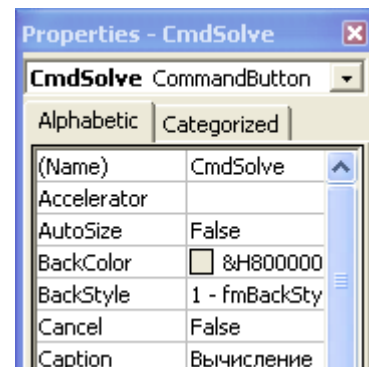
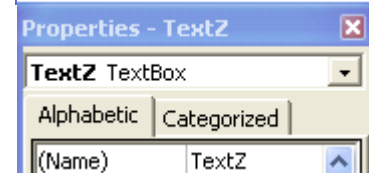
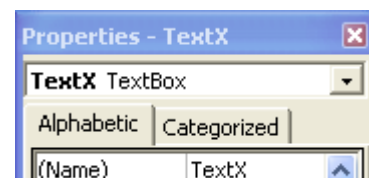
Name – **CmdClr**

Caption – **Очистка**

– для кнопки *Выход*:

Name – **CmdExit**

Caption – **Выход**



ЭТАП 4. Написание обработок событий

В созданной пользовательской форме три кнопки, следовательно, каждой из этих кнопок необходимо написать процедуру обработки ее нажатия. Для написания обработки события «нажатия мышкой» необходимо в режиме добавления элементов управления щелкнуть дважды мышкой по соответствующей кнопке.

Очистка

Для кнопки «Очистка» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdClr_Click()  
  
End Sub
```

2. Написать код процедуры очистки всех элементов управления TextBox (в нашем случае это три элемента – *TextY*, *TextZ* и *TextW*):

```
Private Sub CmdClr_Click()  
    TextY.Text = ""  
    TextZ.Text = ""  
    TextW.Text = ""  
End Sub
```

Выход

Для кнопки «Выход» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdExit_Click()  
  
End Sub
```

2. Написать код процедуры выхода из приложения, что происходит командой End, т.е. код программы примет вид:

```
Private Sub CmdExit_Click()  
    End  
End Sub
```

Для кнопки «Вычисление» необходимо:

Вычисление

1. Произвести двойной щелчок по кнопке и необходимо ввести код процедуры, согласно блок-схеме, т.е. код программы примет вид:

```
Private Sub CmdSolve_Click()  
    Dim g(1 To 6) As Integer  
    Dim a(2 To 5) As Integer  
    Dim k As Integer, i As Integer  
    Dim S As Integer, P As Single  
    Dim y As Integer, z As Single  
    Dim W As Single  
    'Ввод массива gk, k=1..6  
    For k = 1 To 6  
        g(k) = Cells(k + 2, 2)  
    Next  
    'Ввод массива ai, i=2..5  
    For i = 2 To 5  
        a(i) = Cells(i + 1, 5)  
    Next  
    y = CSng(TextY.Text)  
    z = CSng(TextZ.Text)  
    S = 0  
    For k = 1 To 6  
        S = S + g(k)  
    Next  
    P = 1  
    For i = 2 To 5  
        P = P * (a(i) + z * z)  
    Next  
    W = S / y + P / z  
    TextW.Text = CStr(W)  
End Sub
```


Для ввода значений переменных y , z используется функция **CSng()**, которая преобразует текстовое значение в числовое. Циклические процессы реализуются с помощью оператора **For...Next**. Как уже отмечалось ввод значений массивов g_k , a_j происходит с листа Microsoft Excel, поэтому для ввода используется функция **Cells(номер строки, номер столбца)**. В примере для ввода с листа Microsoft Excel в код программы добавлены строки выделенные рамкой. Вывод значения переменной W происходит в текстовое окно с именем **TextW** с помощью функции **CStr()**, которая преобразует числовое значение в текстовое.

В указанной процедуре также необходимо описать все переменные участвующие в программе. Согласно блок-схеме или указанной выше программе, имеем следующие переменные и массивы:

$$k, i, S, P, W, y, z, g_k, a_j.$$

ЭТАП 5. Запуск программы, получение и проверка результатов

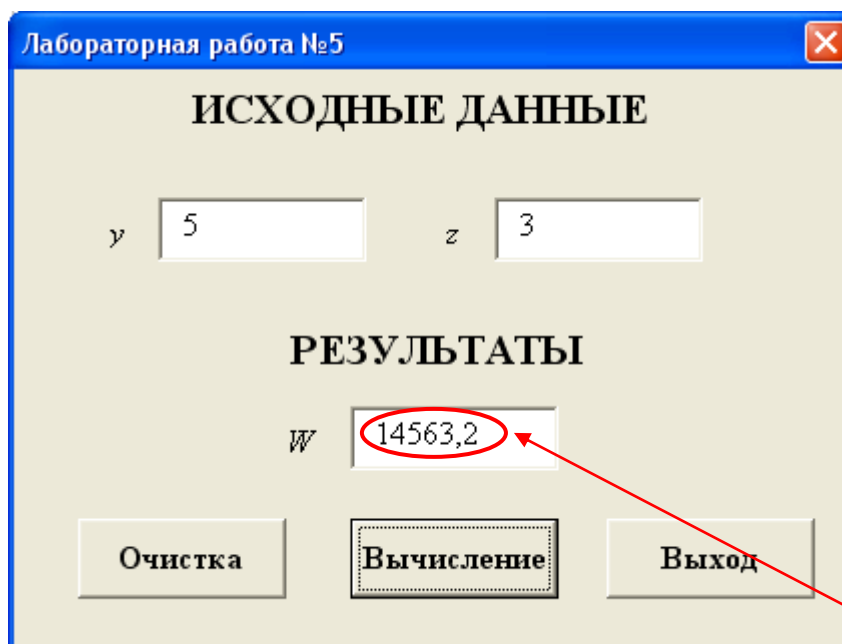
1. Необходимо ввести массивы g_k, a_j как показано на рисунке

2. Для запуска программы необходимо нажать кнопку *Run*. 

3. В появившейся форме заполнить ИСХОДНЫЕ ДАННЫЕ: y, z

	A	B	C	D	E
1	Исходные данные: массивы				
2	k	gk		j	aj
3	1	2		2	5
4	2	4		3	6
5	3	-3		4	7
6	4	5		5	4
7	5	1			
8	6	7			

4. В появившейся форме нажать кнопку *Вычисление*, при этом значение W будет выведено в текстовое поле **TextBox**.



5. Проверка правильности полученных результатов:

$$S = \sum_{k=1}^6 g_k = 2 + 4 + (-3) + 5 + 7 + 1 = 16,$$

$$P = \prod_{j=2}^5 (a_j + z^2) = (5 + 3^2)(6 + 3^2)(7 + 3^2)(4 + 3^2) = 43680,$$

$$\text{тогда } W = \frac{S}{y} + \frac{P}{z} = \frac{16}{5} + \frac{43680}{3} = 14563,2$$

ЛАБОРАТОРНАЯ РАБОТА № 6

Тема: «Организация интерфейса для реализации циклических процессов в среде VBA. Одномерные массивы »

Цель: *Получение навыков составления программ циклических вычислительных процессов. Изучение циклических процессов с использованием одномерных массивов.*

Теоретическая часть

Для описания одномерных массивов в Visual Basic используется конструкция **Dim**. Например, для описания массива целых чисел A_i ($i = \overline{1,10}$) необходимо ввести код

Dim A(1 To 10) As Integer

Для ввода значений в программу с листа Microsoft Excel необходимо использовать функцию Cells. Синтаксис функции

Cells(номер строки, номер столбца)

Например, $x = Cells(3,4)$ вводит в переменную x значение ячейки D3

Пример решения задания

Задание

Задан массив A_i ($i = \overline{1,5}$). Определить и распечатать среднее арифметическое значение всех отрицательных элементов. Значения элементов массива задать самостоятельно и вывести на печать.

Ход решения задания

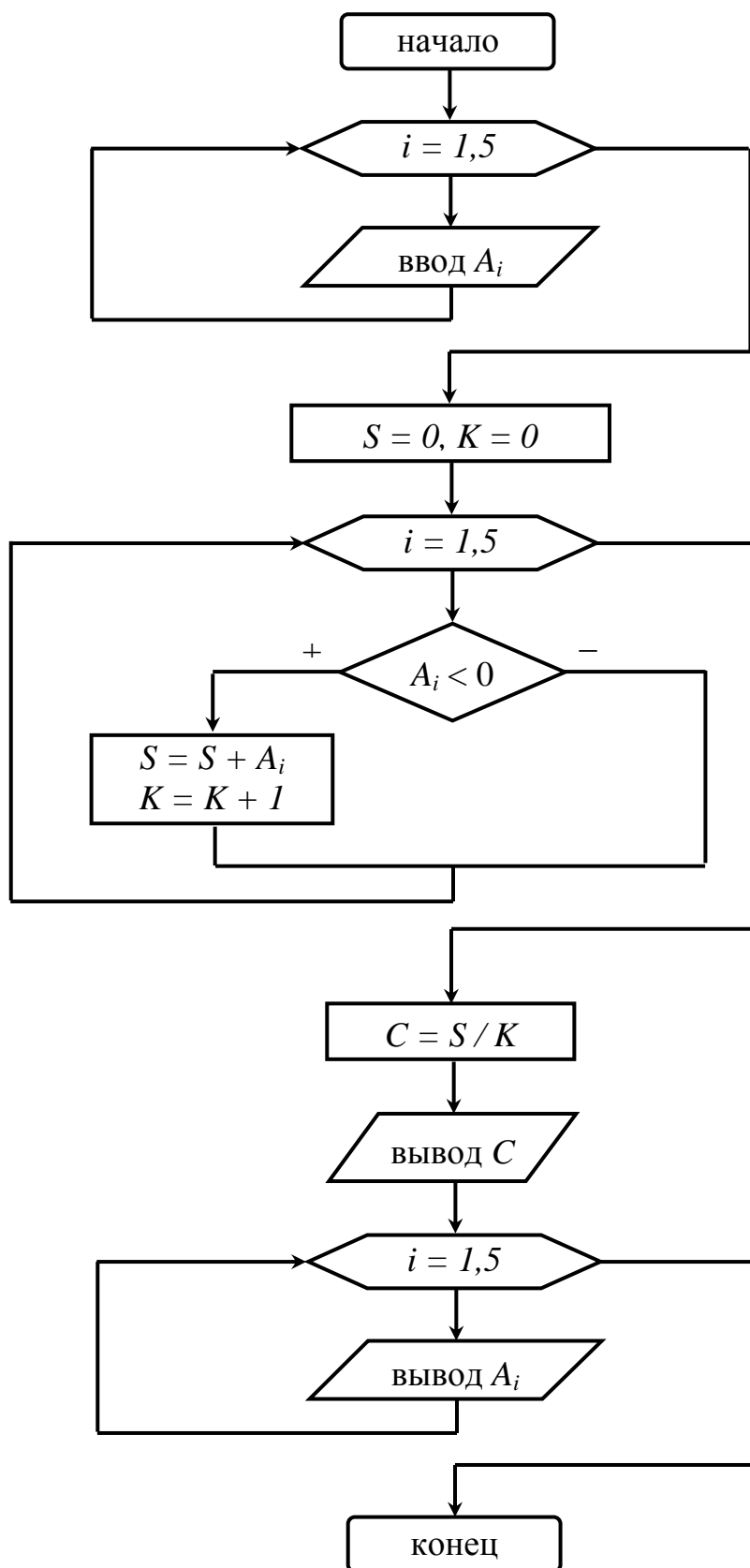
Для решения поставленной задачи необходимо составить математическую модель:

$$C = \frac{S}{K}, \quad S = \sum_{\substack{i=1 \\ A_i < 0}}^5 A_i, \quad K = \sum_{\substack{i=1 \\ A_i < 0}}^5 1$$

где C – среднее арифметическое всех отрицательных элементов;
 S – сумма всех отрицательных элементов;
 K – количество всех отрицательных элементов.

Решение поставленной задачи

ЭТАП 1. Составление блок-схемы

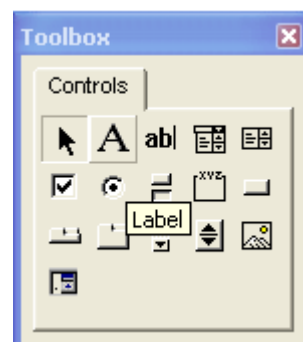


ЭТАП 2. Определение элементов управления ввода-вывода

На данном этапе необходимо определиться с элементами управления, которые будут использоваться для ввода и вывода данных. Добавление элементов происходит с панели инструментов *Toolbox* (в случае отсутствия данной панели на экране необходимо: *Insert* → *Toolbox*)

ИСХОДНЫЕ ДАННЫЕ

В данном случае нет неизвестных данных, требующих ввода, поэтому вводится просто строка подсказка – откуда берутся значения массива, с использованием элемента управления **Label**. Для данных элементов управления свойства можно не менять, т. к. они не будут использоваться в программе.

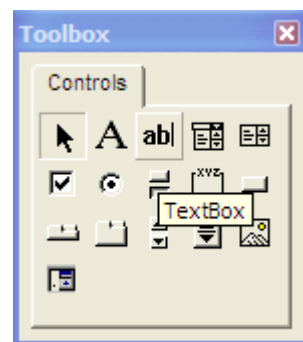


РЕЗУЛЬТАТЫ



Наличие в блок-схеме блока вывода значения переменной *C* требует добавления в форму одного элемента управления типа **TextBox**.

Поэтому создаем для переменной *C* свой элемент управления и меняем некоторые из его свойств:



C – элемент управления **TextBox** с такими свойствами: *Name* – **TextC**

ЭТАП 3. Создание формы с элементами управления

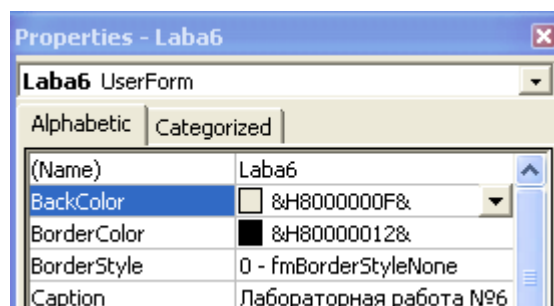


Последовательность создания формы с элементами управления

1. Запускаем программу *Microsoft Excel*
2. Запускаем встроенный редактор VBA: *Сервис* → *Макрос* → *Редактор Visual Basic*
3. Добавляем пользовательскую форму: *Insert* → *UserForm*
4. Задаем свойства формы в окне *Properties* (если данное окно отсутствует, то нужно нажать функциональную клавишу F4):

Name – **Laba6**,

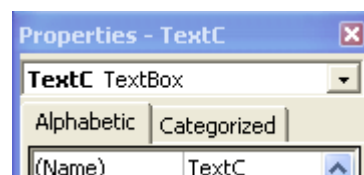
Caption – **Лабораторная работа № 6**



5. Добавляем элементы управления **Label** и **TextBox**, и задаем их свойства. Например,

– для переменной *C*:

Name – **TextC**

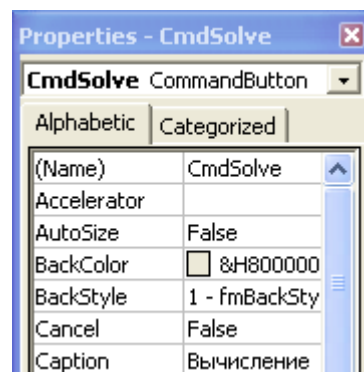


6. Добавляем кнопки **CommandButton** и задаем их свойства. Например,

– для кнопки *Вычисление*:

Name – **CmdSolve**

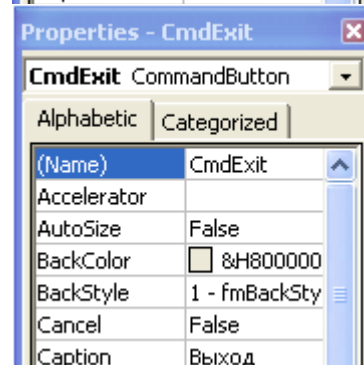
Caption – **Вычисление**



– для кнопки *Очистка*:

Name – **CmdClr**

Caption – **Очистка**



– для кнопки *Выход*:

Name – **CmdExit**

Caption – **Выход**

ЭТАП 4. Написание обработок событий

В созданной пользовательской форме три кнопки, следовательно, к каждой из этих кнопок необходимо написать процедуру обработки ее нажатия. Для написания обработки события «нажатия мышкой» необходимо в режиме добавления элементов управления щелкнуть дважды мышкой по соответствующей кнопке.

Очистка

Для кнопки «Очистка» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdClr_Click()  
  
End Sub
```

2. Написать код процедуры очистки всех элементов управления TextBox (в нашем случае это один элемент – **TextC**):

```
Private Sub CmdClr_Click()  
    TextC.Text = ""  
End Sub
```

Выход

Для кнопки «Выход» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdExit_Click()  
  
End Sub
```

2. Написать код процедуры выхода из приложения, что происходит командой End, т.е. код программы примет вид:

```
Private Sub CmdExit_Click()  
    End  
End Sub
```

Для кнопки «Вычисление» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdSolve_Click()

End Sub
```

2. Написать код процедуры согласно составленной выше блок-схеме, т.е. код программы примет вид:

```
Private Sub CmdSolve_Click()
    Dim A(1 To 5) As Integer
    Dim S As Single, C As Single
    Dim i As Integer, K As Integer
    ' Ввод массива A
    For i = 1 To 5
        A(i) = Cells(i + 2, 2)
    Next
    ' Расчет среднего арифметического
    S = 0: K = 0
    For i = 1 To 5
        If A(i) < 0 Then
            S = S + A(i)
            K = K + 1
        End If
    Next
    C = S / K
    TextC.Text = CStr(C)
    ' Вывод массива A
    For i = 1 To 5
        Cells(i + 2, 4) = A(i)
    Next
End Sub
```

Как уже отмечалось ввод значений массивов A_i происходит с листа Microsoft Excel, поэтому для ввода используется функция **Cells(номер строки, номер столбца)**. Циклические процессы реализуются с помощью оператора **For...Next**. В примере для ввода с листа Microsoft Excel в код программы добавлены строки выделенные рамкой. Вывод значения переменной C происходит в текстовое окно с именем **TextC** с помощью функции **CStr()**, которая преобразует числовое значение в текстовое. Вывод массива A_i происходит на лист Microsoft Excel начиная с ячейки D3 (при $i=1$ имеем `Cells(3, 4)`).

В указанной процедуре также необходимо описать все переменные участвующие в программе. Согласно блок-схеме или указанной выше программе, имеем следующие переменные и массив:

C, S, K, i, A_i .

ЭТАП 5. Запуск программы, получение и проверка результатов

1. Необходимо ввести массив A_i как показано на рисунке (столбцы A и B)

	A	B	C	D
1	<i>Значения элементов массива A_i ($i=1, 5$)</i>			
2	i	A_i		A_i
3	1	-10		-10
4	2	3		3
5	3	-2		-2
6	4	-4		-4
7	5	0		0

2. Для запуска программы необходимо нажать кнопку *Run*.



3. В появившейся форме нажать кнопку *Вычисление*, при этом значение C будет выведено в текстовое поле **TextBox**.

4. Проверка правильности полученных результатов

$$\tilde{N} = \frac{-10 + (-2) + (-4)}{3} = -5,333333$$

ЛАБОРАТОРНАЯ РАБОТА № 7

Тема: «Организация интерфейса для реализации циклических процессов в среде VBA. Двумерные массивы»

Цель: *Получение навыков составления программ циклических вычислительных процессов. Изучение циклических процессов с использованием двумерных массивов.*

Теоретическая часть

Для описания двумерных массивов в Visual Basic используется конструкция **Dim**. Например, для описания массива целых чисел A_{ij} ($i = \overline{1,3}, j = \overline{1,4}$) необходимо ввести код

Dim A(1 To 3, 1 To 4) As Integer

Для ввода значений из программы с листа Microsoft Excel необходимо использовать функцию Cells. Синтаксис функции

Cells(номер строки, номер столбца)

Например, $A(1,2) = Cells(3,4)$ вводит в элемент массива $A(1,2)$ значение ячейки D3

Пример решения задания

Задание

Задан массив A_{ij} ($i = \overline{1,5}; j = \overline{1,6}$). Определить и распечатать среднее арифметическое всех отрицательных элементов. Значения элементов массива задать самостоятельно.

Ход решения задания

Для решения поставленной задачи необходимо составить математическую модель:

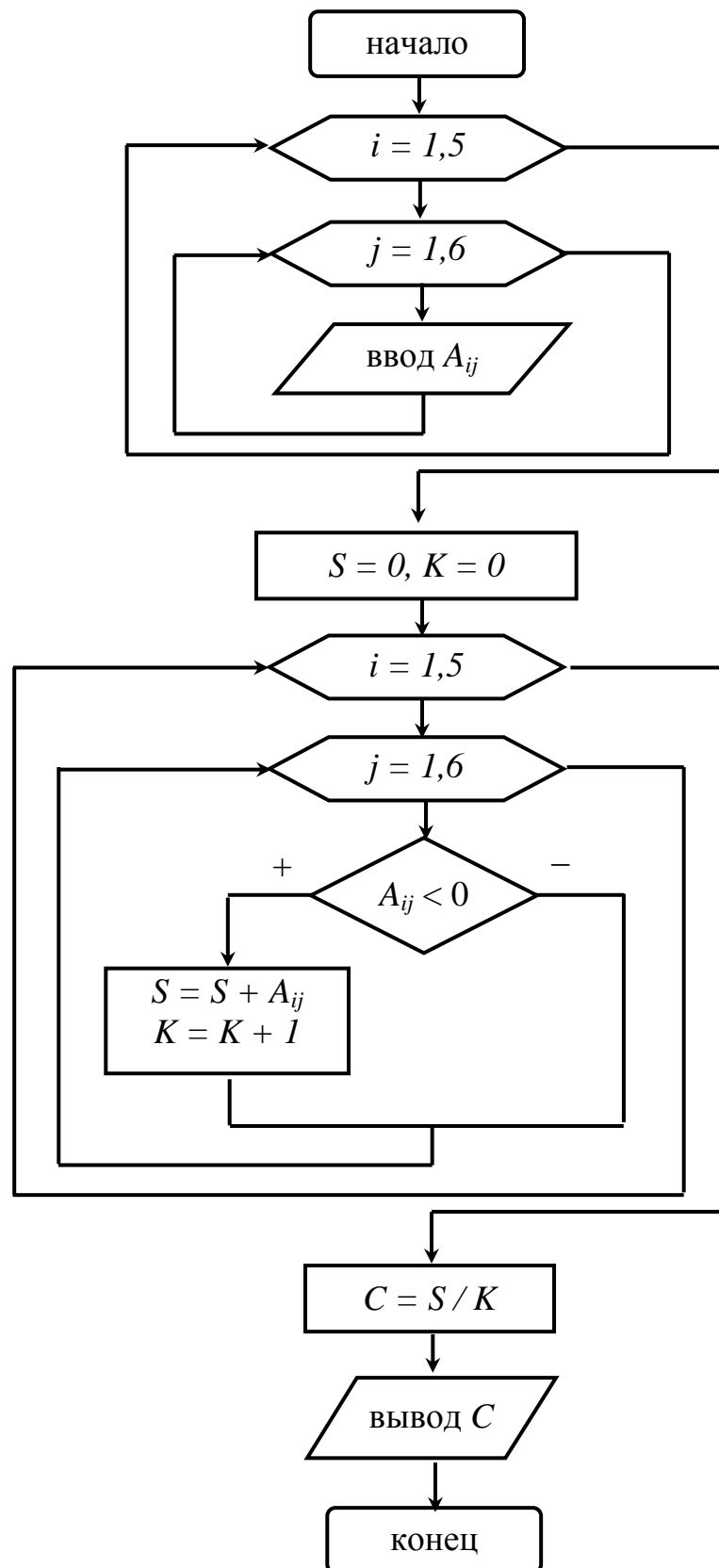
$$C = \frac{S}{K}, \quad S = \sum_{\substack{i=1 \\ A_{ij} < 0}}^5 \sum_{j=1}^6 A_{ij}, \quad K = \sum_{\substack{i=1 \\ A_{ij} < 0}}^5 \sum_{j=1}^6 1$$

где C – среднее арифметическое всех отрицательных элементов;

S – сумма всех отрицательных элементов;

K – количество всех отрицательных элементов.

ЭТАП 1. Составление блок-схемы

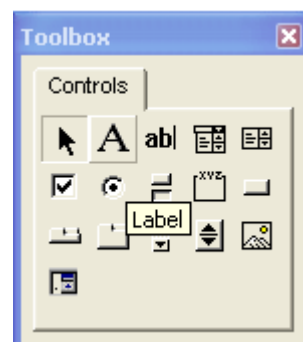


ЭТАП 2. Определение элементов управления ввода-вывода

На данном этапе необходимо определиться с элементами управления, которые будут использоваться для ввода и вывода данных. Добавление элементов происходит с панели инструментов *Toolbox* (в случае отсутствия данной панели на экране необходимо: *Insert* → *Toolbox*)

ИСХОДНЫЕ ДАННЫЕ

В данном случае нет неизвестных данных, требующих ввода, поэтому вводится просто строка подсказка – откуда берутся значения массива, с использованием элемента управления **Label**. Для данных элементов управления свойства можно не менять, т. к. они не будут использоваться в программе.

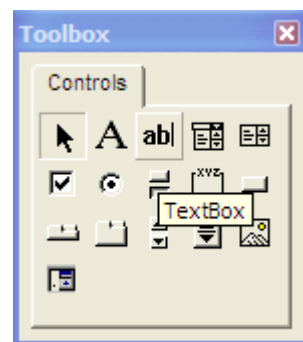


РЕЗУЛЬТАТЫ



Наличие в блок-схеме блока вывода значения переменной *C* требует добавления в форму одного элемента управления типа **TextBox**.

Поэтому создаем для переменной *C* свой элемент управления и меняем некоторые из его свойств:



C – элемент управления **TextBox** с такими свойствами: *Name* – **TextC**

ЭТАП 3. Создание формы с элементами управления

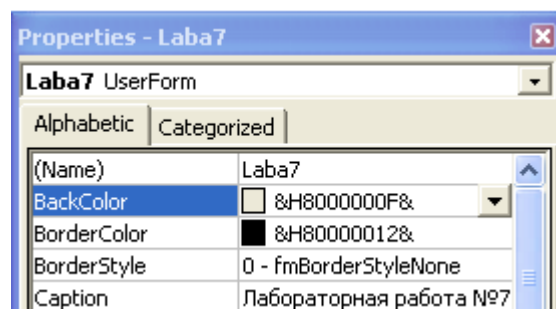


Последовательность создания формы с элементами управления

1. Запускаем программу *Microsoft Excel*
2. Запускаем встроенный редактор VBA: *Сервис* → *Макрос* → *Редактор Visual Basic*
3. Добавляем пользовательскую форму: *Insert* → *UserForm*
4. Задаем свойства формы в окне *Properties* (если данное окно отсутствует, то нужно нажать функциональную клавишу F4):

Name – **Laba7**,

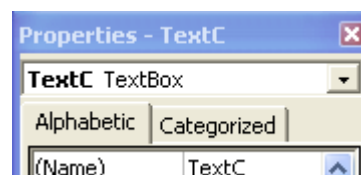
Caption – **Лабораторная работа № 7**



5. Добавляем элементы управления **Label** и **TextBox**, и задаем их свойства. Например,

– для переменной *C*:

Name – **TextC**

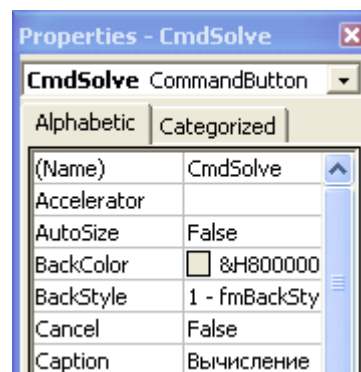


6. Добавляем кнопки **CommandButton** и задаем их свойства. Например,

– для кнопки *Вычисление*:

Name – **CmdSolve**

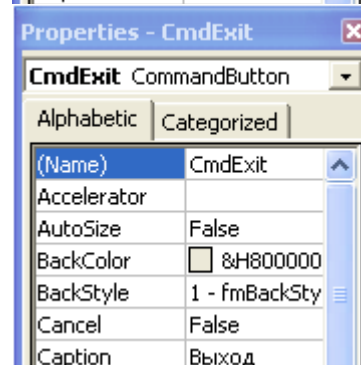
Caption – **Вычисление**



– для кнопки *Очистка*:

Name – **CmdClr**

Caption – **Очистка**



– для кнопки *Выход*:

Name – **CmdExit**

Caption – **Выход**

ЭТАП 4. Написание обработок событий

В созданной пользовательской форме три кнопки, следовательно, к каждой из этих кнопок необходимо написать процедуру обработки ее нажатия. Для написания обработки события «нажатия мышкой» необходимо в режиме добавления элементов управления щелкнуть дважды мышкой по соответствующей кнопке.

Очистка

Для кнопки «Очистка» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdClr_Click()  
  
End Sub
```

2. Написать код процедуры очистки всех элементов управления TextBox (в нашем случае это один элемент – **TextC**):

```
Private Sub CmdClr_Click()  
    TextC.Text = ""  
End Sub
```

Выход

Для кнопки «Выход» необходимо:

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdExit_Click()  
  
End Sub
```

2. Написать код процедуры выхода из приложения, что происходит командой End, т.е. код программы примет вид:

```
Private Sub CmdExit_Click()  
    End  
End Sub
```

Для кнопки «Вычисление» необходимо:

Вычисление

1. Произвести двойной щелчок по кнопке, появится окно Кода формы и в нем будет уже написано:

```
Private Sub CmdSolve_Click()

End Sub
```

2. Написать код процедуры согласно составленной выше блок-схеме, т.е. код программы примет вид:

```
Private Sub CmdSolve_Click()
    Dim A(1 To 5, 1 To 6) As Integer
    Dim S As Single, C As Single
    Dim i As Integer, j As Integer, K As Integer
    ' Ввод массива A
    For i = 1 To 5
        For j = 1 To 6
            A(i, j) = Cells(i + 3, j + 1)
        Next j
    Next i
    ' Расчет среднего арифметического
    S = 0: K = 0
    For i = 1 To 5
        For j = 1 To 6
            If A(i, j) < 0 Then
                S = S + A(i, j)
                K = K + 1
            End If
        Next j
    Next i
    C = S / K
    TextC.Text = CStr(C)
End Sub
```

Как уже отмечалось ввод значений массивов A_{ij} происходит с листа Microsoft Excel, поэтому для ввода используется функция **Cells(номер строки, номер столбца)**. Циклические процессы реализуются с помощью оператора **For...Next**. В примере для ввода с листа Microsoft Excel в код программы добавлены строки выделенные рамкой. Вывод значения переменной C происходит в текстовое окно с именем **TextC** с помощью функции **CStr()**, которая преобразует числовое значение в текстовое.

В указанной процедуре также необходимо описать все переменные участвующие в программе. Согласно блок-схеме или указанной выше программе, имеем следующие переменные и массив:

C, S, K, i, j, A_{ij} .

ЭТАП 5. Запуск программы, получение и проверка результатов

1. Необходимо ввести массив A_{ij} как показано на рисунке

2. Для запуска программы необходимо нажать кнопку *Run*.



	A	B	C	D	E	F	G
1	Значения элементов массива A_{ij} ($i=1,5; j=1,6$)						
2		A_{ij}					
3	ij	1	2	3	4	5	6
4	1	-10	-4	4	6	-10	-10
5	2	3	3	3	3	3	3
6	3	-2	-2	-2	-2	-2	-2
7	4	-4	-4	-4	-4	-4	-4
8	5	0	0	-5	0	0	0

3. В появившейся форме нажать кнопку *Вычисление*, при этом значение C будет выведено в текстовое поле **TextBox**.

4. Проверка правильности полученных результатов

$$\tilde{N} = \frac{-10 - 2 - 4 - 4 - 2 - 4 - 2 - 4 - 5 - 2 - 4 - 10 - 2 - 4 - 10 - 2 - 4}{17} = -4,411765$$

ПРИЛОЖЕНИЕ I.

Соответствие между записью математических выражений и функций в математике и программах

Запись в математике	Запись в программе	Тип результата	Назначение
$x \cdot y$	<code>x*y</code>	-	Произведение чисел
$\frac{x}{y}$	<code>x/y</code>	-	Деление чисел
$\frac{x+a}{y+b}$	<code>(x+a) / (y+b)</code>	-	Сложная дробь
x^2	<code>x*x</code>	-	Квадрат числа
x^3	<code>x*x*x</code>	-	Куб числа
$ x $	<code>abs (x)</code>	real, integer	Модуль аргумента
$\sin x$	<code>sin (x)</code>	real	Синус, угол в радианах
$\cos x$	<code>cos (x)</code>	real	Косинус, угол в радианах
$\arcsin x$	<code>arcsin (x)</code>	real	Арксинус, значение в радианах
x^k	<code>x^k</code>	real	k степень x
\sqrt{x}	<code>sqr (x)</code>	real	Корень квадратный
e^x	<code>exp (x)</code>	real	Экспонента
a^x	<code>exp (x*log (a))</code>	real	Экспонента
$\sqrt[m]{a}$	<code>exp (1/m*log (a))</code>	real	Корень m-ой степени
$\ln x$	<code>log (x)</code>	real	Логарифм натуральный
π	<code>pi</code>	real	$\pi=3.141592653\dots$

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. А. Гарнаев Excel 2000 Разработка приложений, BHV, 2000г.
2. А. Гарнаев Excel, VBA, Internet в экономике, BHV, 2000г.
3. А. Гарнаев Visual Basic 6.0 Разработка приложений, BHV, 2001г.
4. Браун С. Visual Basic 6: учебный курс. – СПб.: Питер, 2002. – 576 с.: ил.
5. Васильев А., Андреев А. VBA в Jffice 2000: учебный курс.-СПб.: Питер, 2001. -432с.
6. Задания к лабораторным работам с методическими указаниями. Раздел. «БЛОК-СХЕМНОЕ ПРОГРАММИРОВАНИЕ. ПРОГРАММИРОВАНИЕ В СРЕДЕ TURBO PASCAL» / Сост.: С.В. Вакуленко, М.Р. Терованесов, Е.И. Демидова, Е.А. Литвинова.
7. Карпов Б. Visual Basic 6: специальный справочник. – СПб.: Питер, 2002. – 416 с.: ил.
8. Кузьменко В.Г. Программирование на VBA 2002. – М.: ООО «Бином-Пресс», 2003 г. – 880 с.; ил.
9. Подлин Ш. Программирование для Microsoft Excel, Вильямс, -М. 2000г.
10. Райтингер М., Муч Г. Visual Basic 6.0: пер. С нем. – К.: Издательская группа BHV, 2000. – 288 с. (Серия «Библиотека студента»)
11. Санна Пол и др. Visual Basic для приложений (версия5) в подлиннике: пер. с англ. – СПб.: BHV – Санкт-Петербург, 1998. – 704 с.
12. Хальворсон М. Visual Basic для профессионалов, ЭКОМ, - М. 2000г.

+

**БАКУЛЕНКО СЕРГЕЙ ВИКТОРОВИЧ,
ТЕРОВАНЕСОВ МИХАИЛ РОМАНОВИЧ,
ДЕМИДОВА ЕЛЕНА ИВАНОВНА,
ЛИТВИНОВА ЕЛЕНА АЛЕКСАНДРОВНА**

**Методические указания
к лабораторным работам
Раздел. «ПРОГРАММИРОВАНИЕ В СРЕДЕ
VISUAL BASIC FOR APPLICATION»**

Ответственная за выпуск Бондаренко Л.В.

**Підписано до друку 20.06.2011.
Формат 60x84/16. Папір Polspeed. Times New Roman.
Друк ксероксний
Умов.друк.арк. 2,7. Наклад 100 прим. Зам. № .**

**Донецький інститут залізничного транспорту
Свідоцтво про внесення до Держ. реєстру від 22.06.2004 р.,
Серія ДК № 1851
Надруковано в редакційно-видавничому відділі ДонІЗТ
83018, м. Донецьк – 18, вул. Горна, 6.**
