



Т.С. Крайнова

ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ

Екатеринбург
2016

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО «УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЛЕСОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра информационных технологий и моделирования

Т.С. Крайнова

ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ

Учебно-методическое пособие
к выполнению лабораторно-практического цикла
для обучающихся по направлениям
09.03.03 «Прикладная информатика», 23.03.03 «Эксплуатация
транспортно-технологических машин и комплексов»,
23.05.01 «Наземные транспортно-технологические средства»
всех форм обучения

Екатеринбург
2016

Печатается по рекомендации методической комиссии ИЭиУ.
Протокол № 2 от 23 сентября 2015 г.

Рецензент – Г.Л. Нохрина, старший преподаватель кафедры ИТиМ.

Редактор А.Л. Ленская
Оператор компьютерной верстки Е.А. Газеева

Подписано в печать 27.04.16		Поз. 85
Плоская печать	Формат 60×84 1/16	Тираж 10 экз.
Заказ №	Печ. л. 1,63	Цена руб. коп.

Редакционно-издательский отдел УГЛТУ
Отдел оперативной полиграфии УГЛТУ

Введение

Visual Basic (VB) является средой разработки, основанной на компонентах и технологиях для создания современных компьютерных приложений. Эта среда эффективно используется при разработке научных, экономических и промышленных проектов.

Слово **Visual** определяет метод, позволяющий создавать то, что пользователь видит на экране, – так называемый **графический интерфейс пользователя**. Слово **Basic** означает язык программирования, прошедший длинный путь развития и в настоящее время являющийся **объектно-ориентированным языком программирования**, основанным на событиях.

События (events) могут создаваться пользователем либо возникать при взаимодействии объектов в процессе выполнения приложения. Типичными событиями являются щелчок мыши (**Click**), нажатие кнопки (**Button_Click**), загрузка формы (**Load**), создание на форме изображений (**Paint**), изменение значения заданной переменной, наступление заданного времени и т.д.

Лабораторная работа № 1

Тема: элементы **InputBox**, **MsgBox**, линейный алгоритм

После запуска Visual Basic (VB) на экране появляется стартовая страница, показанная на рис. 1.

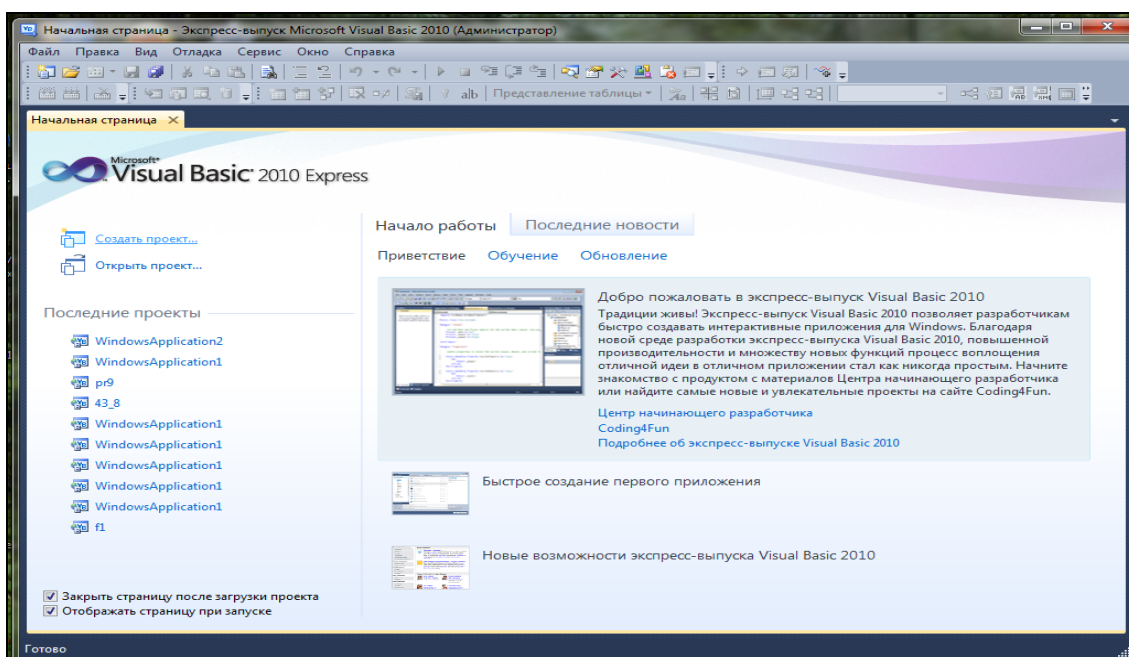


Рис. 1. Стартовая страница VB

Чтобы создать новый проект, необходимо щелкнуть по ссылке Создать проект... и в раскрывающемся списке выбрать Приложение Windows Forms. В нижней строке этого окна в строке Имя проекта (по умолчанию проекту присваивается имя WindowsApplication1) присвоить имя и щелкнуть по кнопке Ok.

В VB программа разработки называется **проектом**. Файл включает проект (.vbprog) и файл решения (.sln). Файл проекта содержит сведения об одном или нескольких проектах.

Создание любого проекта начинается с разработки дизайна формы – **интерфейса пользовательского приложения**, который позволяет пользователю вводить данные, видеть происходящие события и получать результаты. На форме можно печатать текст, рисовать фигуры и размещать **элементы управления (Controls)**, с которыми пользователь может взаимодействовать.

После входа в среду разработки на экране появляется **дизайнер форм (Designer)**, представленный на рис. 2.

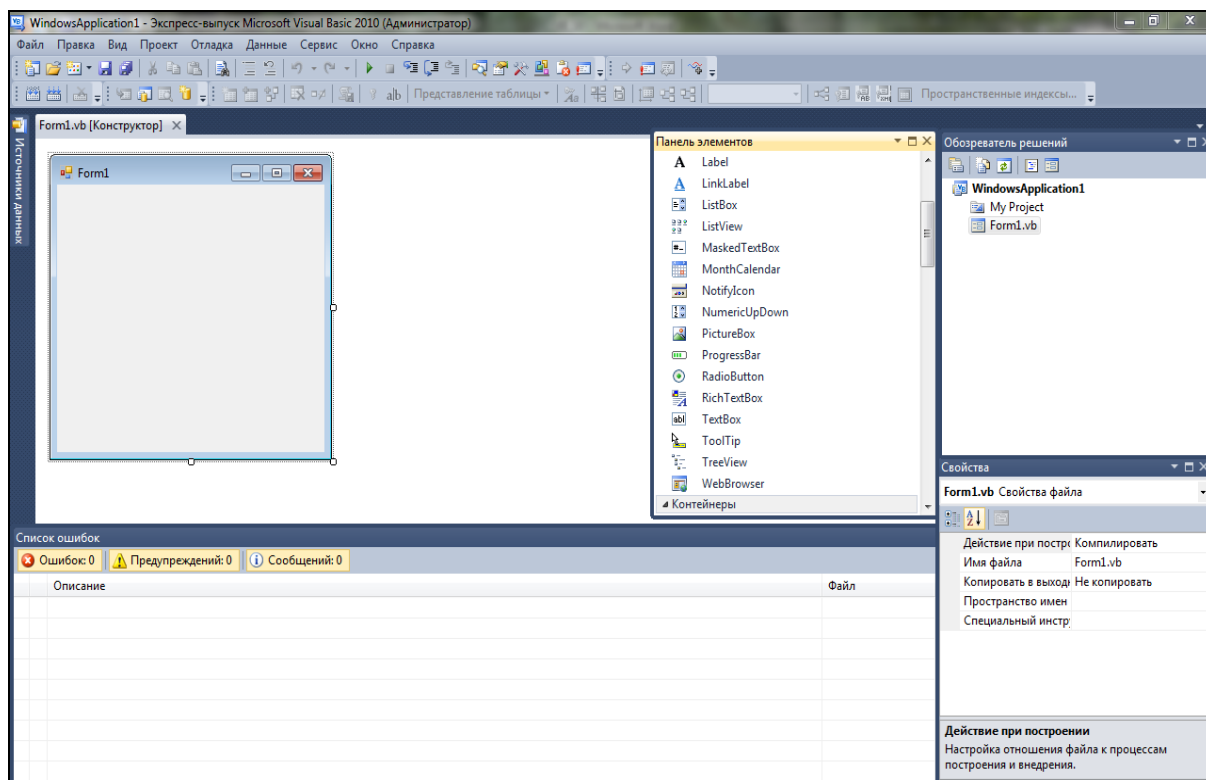


Рис. 2. Вид окна разработки приложения VB

Окна, из которых состоит Designer, можно перемещать в разные места экрана, изменять их размеры.

Создание графического интерфейса проекта начинается с добавления элементов управления на форму с **Панели элементов (Ctrl+Alt+X)**. Элементы перетаскиваются на форму левой кнопкой мыши.

Например:

Label – отображение на форме текста;

TextBox – отображение данных;

Button – командная кнопка.

Каждый элемент управления имеет набор свойств в зависимости от действий, которые с ним связаны. Используя **Окно свойств** (properties window), можно изменять такие свойства элементов, как цвет, размер, имя.

Для **создания события** на любой элемент управления необходимо дважды щелкнуть левой кнопкой мыши по этому элементу. После выполненных действий откроется окно **Редактора кода** (code editor). Там уже будет некоторый код, написанный самой системой, курсор установится между строками, задающими начало и конец процедуры, где и нужно печатать требуемый код. Например, на рис. 3 представлен пример автоматически сформированного кода для события - нажатие по кнопке **Button1**.

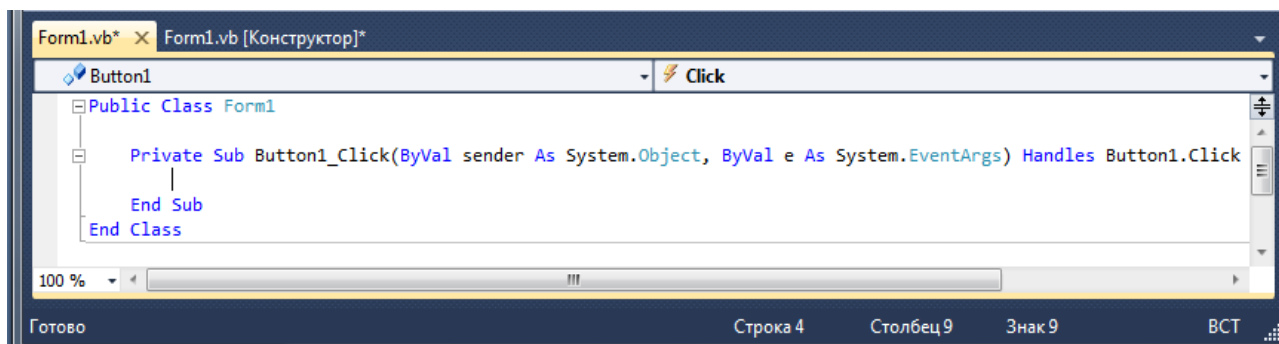



Рис. 3. Обработчик события Нажатие по кнопке

Запуск приложения выполняется либо нажатием функциональной клавиши **F5**, либо меню **Отладка – Начать отладку**, либо щелкнуть по кнопке  на панели инструментов среды **VB**.

Для **сохранения проекта** необходимо:

- 1) создать на носителе информации папку (проект содержит несколько файлов);
- 2) выбрать в меню **Файл – Сохранить все**:
 - а) в строке **Имя** – указать имя создаваемого проекта;
 - б) в строке **Расположение** – указать адрес созданной папки, используя кнопку **Обзор**;
- 3) должен быть установлен маркер **Создать каталог для решения**;
- 4) нажать кнопку **Сохранить**.

Типы данных **VB**

Создание программного кода должно начинаться с определения типов всех используемых переменных (табл. 1).

Типы данных

Тип	Наименование	Диапазон значений
Byte	байт	целое число от 0 до 255
Boolean	логический	<i>true</i> (истина) или <i>false</i> (ложь)
Integer	целое	от -32768 до 32767
Long	длинное целое	от -2 147 483 648 до 2 147 483 647
Single	с плавающей точкой одинарной точности	для отрицательных чисел: от -3.402823E+38 до -1.401298E-45 для положительных чисел: от 1.401298E-45 до 3.402823E+38
Double	с плавающей точкой двойной точности	для отрицательных чисел: от -1.79769313486232E+308 до -4.94065645841247E-324 для положительных чисел: от 4.94065645841247E-324 до 1.79769313486232E+308
Currency	денежный	от -922 337 203 685 477.5808 до 922 337 203 685 477.5807
Date	дата	от 01.01.100 до 31.12.9999
String	строка	приблизительно 2 млрд. символов
Variant	вариантный	любое число или любая строка
User-defined	любой	определяется в соответствии с заданным типом данных

Переменные используются для представления данных любой информации (текстовой, числовой, символьной и т.д.). Для объявления переменной используется ключевое слово **Dim**. Например:

```
Dim A as Integer
Dim B as String
```

Типы переменных можно не объявлять, для этого в начале программного кода проекта надо установить для опции компилятора **Option Explicit** значение **Off**.

Option Explicit Off

Повысить надежность программы может опция **Option Strict**, при которой компилятор проверяет, все ли значения, присваиваемые переменным, соответствуют типам, объявленным для этих переменных.

Option Strict On

Для преобразования примитивных типов данных используются методы, представленные в табл. 2.

Таблица 2

Основные методы преобразования типов данных

Метод	Описание
Val	текст – в число
Str	число – в текст
CStr	любой тип – в строчный
CInt	любой тип – в целый
CSng	любой тип – в вещественный
CDBl	любой тип – в 64-разрядное

Например:

- преобразование числовой переменной **A** в символьную для последующей ее записи в текстовом окне:

```
TextBox1.Text = Str(A)
```

- переменная вещественного типа **B** может быть преобразована в целый тип **Integer**:

```
Dim B as Single
```

```
Dim D as Integer
```

```
D=CInt(B)
```

Математические методы и функции VB

При использовании любой математической функции необходимо в редакторе кода формы поместить строку:

```
Imports System.Math
```

Основные математические методы представлены в табл. 3.

Таблица 3

Математические функции

Метод	Описание
Abs(n)	модуль числа
Atan(n)	арктангенс (в радианах)
Log(n)	натуральный логарифм
Pow(n,m)	n^m
RND(n)	случайное число на отрезке [0,1]
Cos(n)	косинус угла (в радианах)
Exp(n)	e^x
Sin(n)	синус угла (в радианах)
Sqrt(n)	квадратный корень
Tan(n)	тангенс угла (в радианах)

Visual Basic (VB) позволяет выполнять любые арифметические действия с переменными в полном соответствии правилам арифметики (табл. 4).

Основные арифметические операторы

Оператор	Описание
+	сложение
-	вычитание
*	умножение
/	деление
\	целочисленное деление (без остатка)
^	возведение в степень
mod	остаток от деления по модулю
&	сцепление строк (конкатенация)
<>	Не равно

Для слияния двух строк в одну используется оператор *конкатенации* (знак “ & ”). Результат этой операции всегда имеет тип **String**.

Например,

FIO = “Иванов” & “ ” & “Иван” & “ ” & “Иванович”

Результат – в переменную FIO будет помещена строка **Иванов Иван Иванович**.

InputBox

Встроенная функция **InputBox()** высвечивает диалоговое окно для ввода значения. Имеет текстовый тип хранимых данных.

Например,

Dim A as Integer

A=Val(InputBox(“Введите начальное значение переменной A”))

После нажатия по кнопке получим результат (рис. 4).

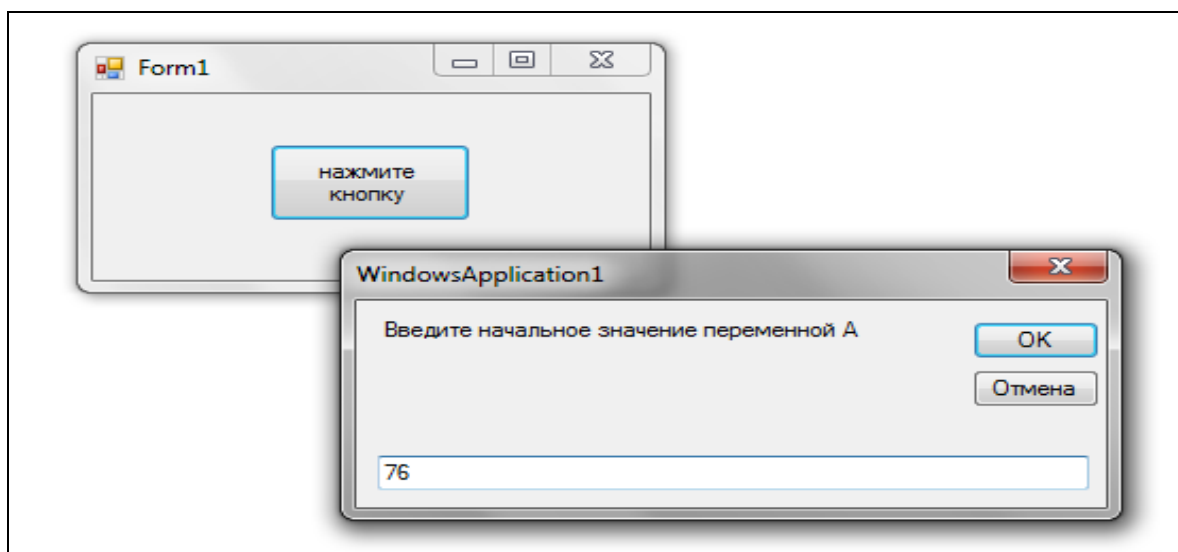


Рис. 4. Результат нажатия по кнопке, метод InputBox

MsgBox

Встроенная функция MsgBox используется для вывода сообщений. Текст сообщения является строкой символов.

Например,

```
Dim B as String
```

```
B=MsgBox ("Доброе утро")
```

После нажатия по кнопке получим результат (рис. 5).

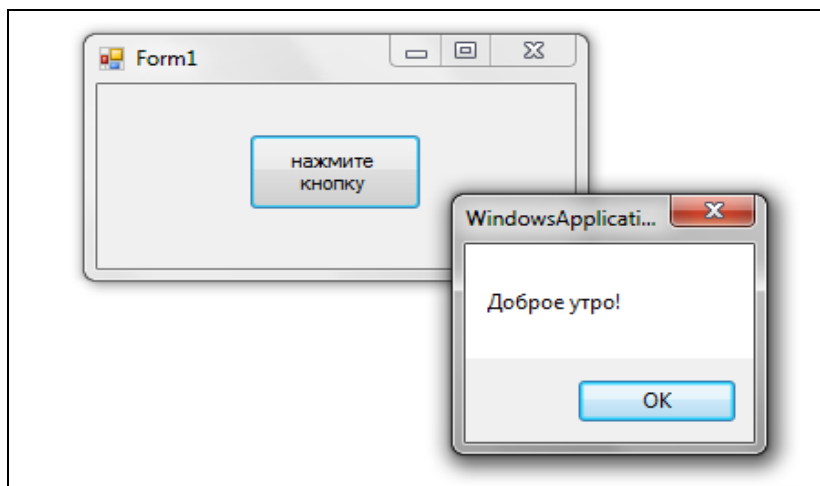


Рис. 5. Результат нажатия по кнопке, метод MsgBox

Задание

Создать приложение, рассчитывающее значение переменной $c = a + b$. Переменные a и b задаются в TextBox.

1. Создать папку Lab1.

2. На форму добавить элементы (рис. 6) и задать свойства (табл. 5).

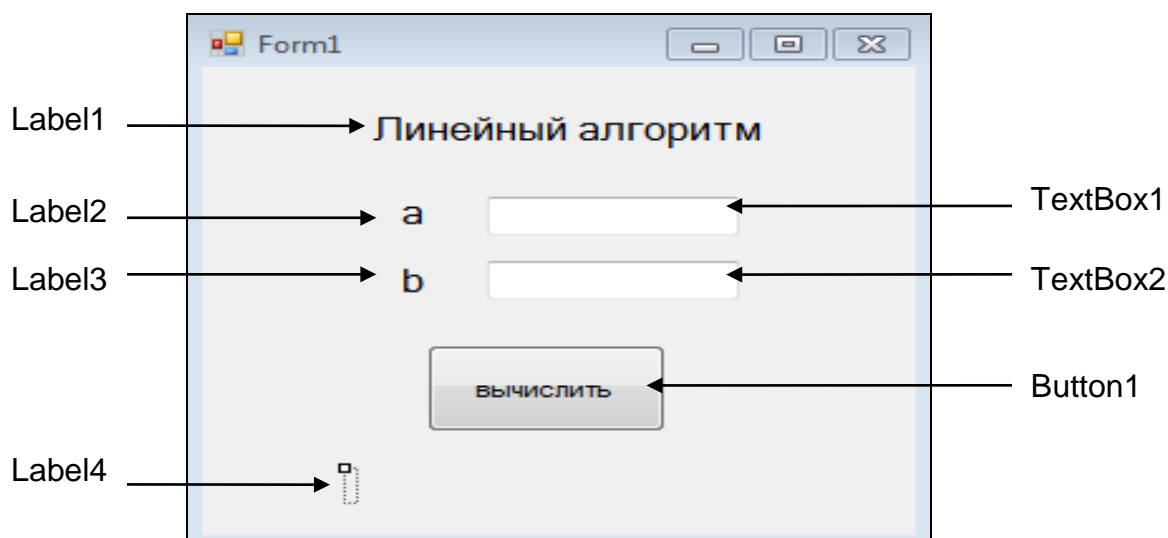


Рис. 6. Вид формы Линейный алгоритм

Свойства элементов

Элемент	Свойство	Значение
Label1	Text	Линейный алгоритм
Label2	Text	a
Label3	Text	b
Label4	Text	
TextBox1	Text	
TextBox2	Text	
Button1	Text	Вычислить

3. Двойным щелчком по элементу **Button1** открыть окно редактора кода и ввести требуемый программный код:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
Dim a, b, c As Double
```

```
a = TextBox1.Text
```

```
b = TextBox2.Text
```

```
c = a + b
```

```
Label4.Text = "a + b =" + Str(c)
```

```
End Sub
```

4. Сохранить проект.

5. Проверить работу проекта.

Дополнительные задания

1. Выбрать вариант индивидуального задания.

2. Вычислить двумя способами, нарисовать блок-схему решения задачи:

а) ввод значений через элемент **TextBox**, ответ вывести в элементе **Label**;

б) ввод значений через диалоговое окно **InputBox**, ответ вывести в элементе **MsgBox**.

Лабораторная работа № 2

Тема: условный оператор **IF..Then..Else**

Структурой ветвления называется такая структура, которая обеспечивает выбор из двух направлений реализуемого процесса. Выбор зависит от условия, имеющегося в данной структуре.

Оператор **If** организует выполнение блока операторов, если заданное условие выполняется, и не делает ничего, если заданное условие не выполняется. Если оператор **If** простой, то **Else** не записывается.

В общем виде:

а) строчный оператор:

```
If условие Then значение_если_истина Else значение_если_ложь
```

б) блочный оператор:

```
If условие Then
    (Операторы 1)
Else
    (Операторы 2)
End If
```

Условие всегда содержит знаки сравнения: больше, меньше, равно и т.д.

Если выполняемое *условие* **ИСТИННО** (True), выполняется *блок операторов*, следующий за словом **Then**, если значение **ЛОЖНО** (False), выполняется *блок операторов*, следующий за словом **Else**. **End If** – завершающий оператор.

Над условными выражениями допустимы операции алгебры логики:

AND (и) – возвращает значение **ИСТИНА**, если все операнды, участвующие в операции, имеют значение **ИСТИНА**, и значение **ЛОЖЬ** во всех остальных случаях.

OR (или) – возвращает значение **ИСТИНА**, если хотя бы один из операндов, участвующих в операции, имеет значение **ИСТИНА**, и значение **ЛОЖЬ**, только когда оба операнда имеют значение **ЛОЖЬ**.

NOT (не) – операция отрицания. Если значение выражения **ИСТИННО**, то возвращается **ЛОЖЬ** и наоборот.

Например, вычислить значение переменной **c**:

```
If a > b then c = a * b else c = a + b
```

Варианты решения:

1) $a = 3, b = 2$

```
If 3 > 2 Then c = 3 * 2 else c = 3 + 2
```

Поскольку выполняемое условие ($3 > 2$) **ИСТИННО** (True), ответом будет блок операторов, стоящий после **Then**, в ответе $c = 3 * 2, c = 6$.

2) $a = 4, b = 6$

```
If 4 > 6 Then c = 4 * 6 else c = 4 + 6
```

Поскольку выполняемое условие ($4 > 6$) **ЛОЖНО** (False), ответом будет блок операторов, стоящий после **else**, $c = 4 + 6, c = 10$.

Задание

Разработать приложение, рассчитывающее значение переменной c :

$$c = \begin{cases} ab & \text{при } a \geq b, a \neq b \\ \frac{a}{b} & \text{при } a < b, b \neq 0 \end{cases}$$

1. Создать папку **Lab2**.
2. Добавить на форму элементы (рис. 7) и задать свойства (табл. 6).

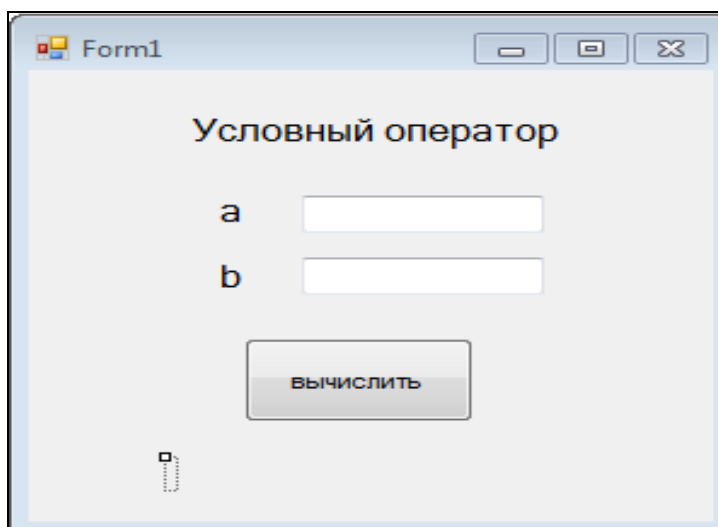


Рис. 7. Вид формы Условный оператор

Таблица 6

Свойства элементов

Элемент	Свойство	Значение
TextBox1	Text	
TextBox2	Text	
Label1	Text	Условный оператор
Label2	Text	a
Label3	Text	b
Label4	Text	
Button1	Text	Вычислить

3. Написать программный код для элемента **Button1**:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim a, b, c As Double
```

```
a = TextBox1.Text
b = TextBox2.Text
If (a > b) Or (a = b) And (a <> 0) Then c = a * b
If (a < b) And (b <> 0) Then c = a / b
Label4.Text = "c = " + Str(c)
End Sub
4. Сохранить проект.
5. Проверить работу приложения.
```

Дополнительные задания

1. Выбрать вариант индивидуального задания.
2. Вычислить двумя способами, нарисовать блок-схему решения задачи:
 - а) ввод значений через элемент `TextBox`, ответ вывести в элементе `Label`;
 - б) ввод значений через диалоговое окно `InputDialog`, ответ вывести в элементе `MsgBox`.

Лабораторная работа № 3

Тема: циклы **For..Next**, **Do While..Loop**, **Do..Loop Until**

Циклом называется алгоритм, который обеспечивает повторение каких-либо действий.

Циклы бывают трех видов: с предусловием, с постусловием, с параметром.

Цикл с параметром используется, когда число повторений цикла известно до начала его работы. При создании такого цикла надо определить его переменную, которая служит *счетчиком* числа повторений цикла, и задать ее *начальное* и *конечное значения*. В качестве дополнительного параметра можно указывать величину изменения счетчика – *шаг (Step)*, который не является обязательным параметром. Если *шаг* равен 1, то его можно не писать.

В общем виде:

```
For счетчик=нач_значение TO кон_значение STEP шаг изменения
    операторы_цикла
Next счетчик
```

При программировании иногда сложно сказать заранее, сколько повторяющихся действий необходимо выполнить. Чтобы добиться заданной точности, используются циклы, управляемые условием. Существует два основных типа условных циклов:

1) **цикл с предусловием Do While...Loop** выполняется до тех пор, пока условие истинно; операторы цикла могут не выполняться ни разу, если его условие окажется ложным при входе в цикл.

В общем виде:

```
счетчик=нач_значение
Do While условие
операторы_цикла
Loop
```

2) **цикл с постусловием Do..Loop Until** указывает, что цикл выполняется, пока условие ложно; операторы цикла всегда будут выполнены, по крайней мере, один раз.

В общем виде:

```
счетчик=нач_значение
Do
операторы_цикла
Loop Until условие
```

Задание

Задача 1. Создать приложение, рассчитывающее сумму последовательных элементов массива А [1,2,3,4,5]. Использовать элемент InputBox для задания начального и конечного элементов массива.

На одном примере рассмотрим применение всех трех типов циклов для решения поставленной задачи.

1. Создать папку Lab3.1.
2. Добавить на форму элементы (рис. 8) и задать свойства (табл. 7).

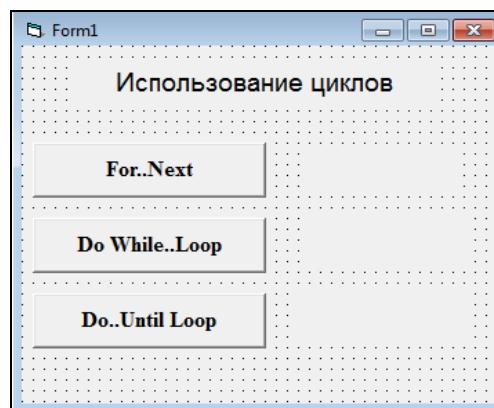


Рис. 8. Вид формы Использование циклов

Свойства элементов

Элемент	Свойство	Значение
Label1	Text	Использование циклов
Label2	Text	
Label3	Text	
Label4	Text	
Button1	Text	For..Next
Button2	Text	Do While..Loop
Button3	Text	Do..Until Loop

3. Написать программный код для элемента **Button1**:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim i, Sum, nach_zn, kon_zn As Integer
    nach_zn = Val(InputBox("Введите начальное значение массива"))
    kon_zn = Val(InputBox("Введите конечное значение массива"))
    Sum = 0
    For i = nach_zn To kon_zn
        Sum = Sum + i
    Next i
    Label2.Text = Str(Sum)
End Sub
```

4. Написать программный код для элемента **Button2**:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim i, Sum, nach_zn, kon_zn As Integer
    nach_zn = Val(InputBox("Введите начальное значение массива"))
    kon_zn = Val(InputBox("Введите конечное значение массива"))
    Sum = 0
    i = nach_zn
    Do While i <= kon_zn
        Sum = Sum + i
        i = i + 1
    Loop
    Label3.Text = Str(Sum)
End Sub
```

5. Написать программный код для элемента **Button3**:

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Dim i, Sum, nach_zn, kon_zn As Integer
    nach_zn = Val(InputBox("Введите начальное значение массива"))
```



```

kon_zn = Val(InputBox("Введите конечное значение массива"))
Sum = 0
i = nach_zn
Do
    Sum = Sum + i
    i = i + 1
Loop Until i > kon_zn
Label4.Text = Str(Sum)
End Sub
    
```

6. Сохранить проект.
7. Проверить работу проекта.

Задача 2. Создать приложение, рассчитывающее количество четных последовательных элементов массива P [-5..5]. Использовать элемент Label для задания начального и конечного элементов массива.

1. Создать папку Lab3.2.
2. Добавить на форму элементы (рис. 9) и задать свойства (табл. 8).

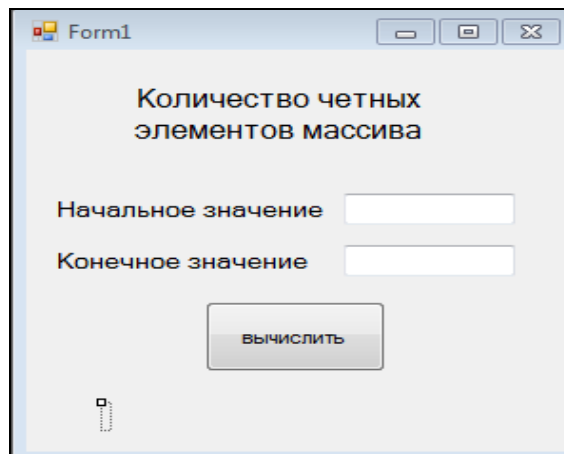


Рис. 9. Вид формы Количество четных элементов массива

Таблица 8

Свойства элементов

Элемент	Свойство	Значение
Label1	Text	Количество четных элементов массива
Label2	Text	Начальное значение
Label3	Text	Конечное значение
Label4	Text	
TextBox1	Text	
TextBox2	Text	
Button1	Text	Вычислить

3. Написать программный код для элемента **Button1**:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim i, Kol, nach_zn, kon_zn As Single
    nach_zn = CSng(TextBox1.Text)
    kon_zn = CSng(TextBox2.Text)
    Kol = 0
    For i = nach_zn To kon_zn
        If (i Mod 2 = 0) And (i <> 0) Then Kol = Kol + 1
    Next i
    Label4.Text = "Ответ = " + Str(Kol)
End Sub
```

4. Сохранить проект.

5. Проверить работу проекта.

Дополнительные задания

1. Написать программный код, позволяющий вычислить произведение и среднее последовательных элементов массива **K**. Начальное и конечное значения массива задать элементом **InputBox**. Ответы вывести в соответствующие элементы **Label**.

2. Написать программный код, рассчитывающий количество кратных трём последовательных элементов массива **F** [-129..817]. Начальное и конечное значения массива задать элементом **TextBox**. Ответ вывести в **MsgBox**.

3. Создать проект, позволяющий вычислить сумму отрицательных нечетных последовательных элементов массива **L**.

Лабораторная работа № 4

Тема: элемент управления **ListBox**

В языке **VB** для наглядного представления информации используется элемент управления типа список – **ListBox** (рис. 10).

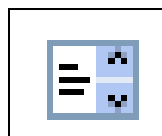


Рис. 10. Пиктограмма элемента **ListBox**

ListBox не только позволяет просматривать элементы, у него также автоматически появляется полоса прокрутки, если элементов в списке оказывается больше, чем можно разместить на экране. Полоса прокрутки позволяет перемещать элементы вверх или вниз.

При размещении первого элемента управления **ListBox** на форме появляется пустой прямоугольник, внутри которого находится стандартный текст.

Для того, чтобы занести в список новый элемент, используется метод **Items.Add()**.

Например, занести значение переменной **X** в список с именем **ListBox1**:

```
ListBox1.Items.Add(x)
```

Например, необходимо занести две переменные **X** и **Y**, это можно сделать так:

```
ListBox1.Items.Add("X= " & X & "Y=" & Y)
```

Задание

Создать приложение, которое позволит вывести элементы массива **A** [16..57], кратные трём, в элемент **ListBox**. Использовать элементы **TextBox** для задания начального и конечного элементов массива.

1. Создать папку **Lab4**.
2. Добавить на форму элементы (рис. 11) и задать свойства (табл. 9).

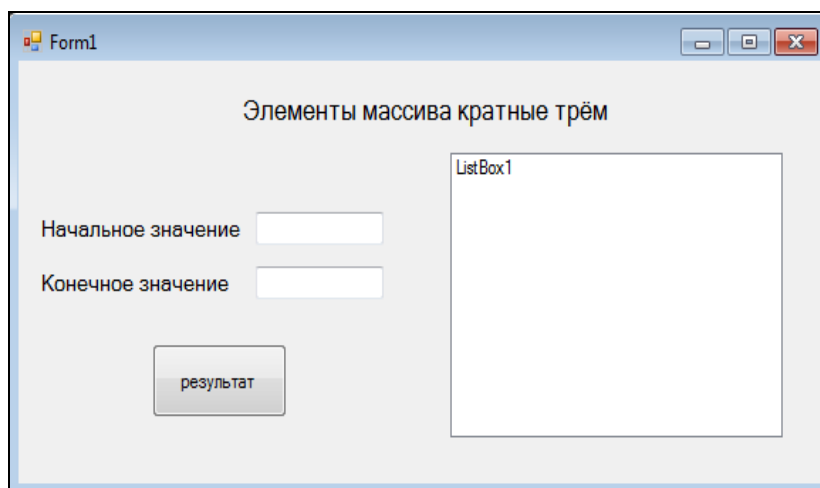


Рис. 11. Вид формы Элементы массива, кратные трём

Таблица 9

Свойства элементов

Элемент	Свойство	Значение
Label1	Text	Элементы массива, кратные трём
Label2	Text	Начальное значение
Label3	Text	Конечное значение
TextBox1	Text	
TextBox2	Text	
Button1	Text	Результат

3. Написать программный код для элемента Button1:
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Dim nach_zn, kon_zn, i As Integer
nach_zn = Val(TextBox1.Text)
kon_zn = Val(TextBox2.Text)
For i = nach_zn To kon_zn
If i Mod 3 = 0 Then ListBox1.Items.Add(i)
Next i
End Sub

4. Сохранить проект.
5. Проверить работу приложения.

Дополнительные задания

1. Вывести таблицу значений функции $y=x+e^x$ на отрезке $[-2..2]$, шаг $h=0,1$.
2. Вывести таблицу значений на отрезке $[-2..6]$, шаг $h=0,5$ функции:

$$y = \begin{cases} 5 * e^x, & \text{если } 0 \leq x \leq 5 \\ 2 * \sin(x), & \text{если } x > 5 \\ |x|, & \text{если } x < 0 \end{cases}$$

3. Вывести таблицу и вычислить сумму значений функции на отрезке $[1,9..2,1]$, шаг $h=0,01$:

$$y = x^4 + 39 * x^3 + 958 * x^2 + 1081 * x - 1987$$

Лабораторная работа № 5

Тема: одномерные массивы

Массив – это упорядоченная совокупность элементов одного типа, имеющих общее имя. Для обращения к элементам массива необходимо указать имя и индекс требуемого элемента. Нумерация элементов массива начинается с нуля.

Для работы с массивами их необходимо объявить. Если массив имеет фиксированную длину, то надо указать его имя, число элементов и тип.

Например, массив целых чисел **B** состоит из пяти элементов:

```
Dim B(4) As Integer
```

При объявлении массива с небольшим числом элементов может быть выполнена и его инициация. При этом в скобках не указывается количество элементов массива, поскольку оно вычисляется автоматически по количеству заданных значений. Задаваемые значения заключаются в фигурные скобки.

Например, описать заданный массив $x(2)$ чисел с плавающей точкой одинарной точности:

```
Dim x() As Single = {5.6, 7.23, 18.4}
```

После этого объявления элементы получают следующие значения:

```
x(0) = 5.6
```

```
x(1) = 7.23
```

```
x(2) = 18.4
```

Размер динамического одномерного массива можно задавать во время работы программы с помощью оператора **ReDim**.

В общем виде:

```
ReDim имя_массива (размер)
```

Сначала необходимо объявить массив, указав его имя и тип, не указывая число элементов массива. Затем надо определить размер массива, используя оператор **InputBox**.

Например, задать размер одномерного массива последовательных целых чисел P , n – число элементов массива:

```
Dim P() As Integer
```

```
Dim n As Integer
```

```
n = Val(InputBox("Введите размер массива"))
```

```
ReDim P(n)
```

Значения элементов массива, определяемые пользователем, **всегда задаются циклом**.

Например, задать и заполнить произвольными значениями одномерный массив Q , где n – количество элементов, $e1$ – значение элемента массива:

```
Dim Q() As Integer
```

```
Dim n, e1 As Integer
```

```
n = Val(InputBox("Введите размер массива"))
```

```
ReDim Q(n)
```

```
For i = 0 To n
```

```
    e1 = Val(InputBox("Введите " & i & " элемент массива Q"))
```

```
    Q(i) = e1
```

```
Next i
```

Оператор **Erase имя_массива** позволяет очистить массив от элементов в памяти целиком. Например, очистить содержимое массива A и заполнить его новыми значениями:

```
Erase A
```

Оператор **UBound (имя_массива)** позволяет программно определить размерность массива при выполнении любых действий над элементами массива. Например, найти количество положительных элементов массива $A(4)$ с номерами элементов, кратных 3:

```
Kol = 0
```

```
For i = 0 To UBound(A)
```

```
    If (i mod 3 = 0) and (A(i)>0) then Kol = Kol + 1
```

```
Next i
```

Задание

Задача 1. Создать приложение, рассчитывающее количество элементов массива $A [n]$, n – количество элементов массива с номерами элементов, кратных 3. Использовать `InputBox` для задания размера и значений элементов массива. Вывести элементы массива в `Label`.

1. Создать папку `Lab51`.
2. Добавить на форму элементы (рис. 12) и задать свойства (табл. 10).

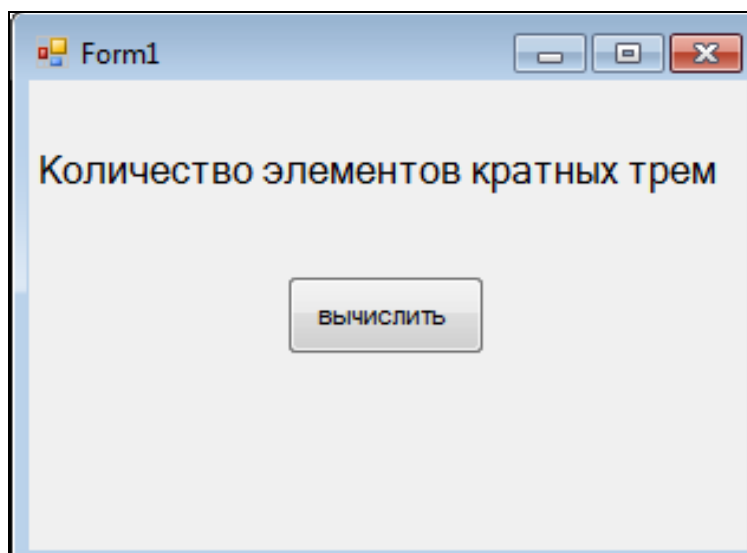


Рис. 12. Вид формы Количество элементов, кратных трем

Таблица 10

Свойства элементов

Элемент	Свойство	Значение
Label1	Text	Количество элементов, кратных трем
Label2	Text	
Label3	Text	
Button1	Text	Вычислить

3. Для элемента `Button1` написать программный код:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim A(), n, Kol, el, i As Integer
    'очистить от данных Label2
    Label2.Text = " "
    'здать количество элементов в массиве
    n = Val(InputBox("Введите число элементов массива»))
    'количество элементов массива A будет равно введенному n
    ReDim A(n)
```

```

' заполнить массив A элементами через InputBox
For I = 0 To n
    e1 = Val(InputBox("Введите " & I & " элемент массива A"))
    A(i) = e1
Next i
' вывести элементы массива A в Label2
For I = 0 To Ubound(A)
    Label2.Text = Label2.Text & " " & CStr(A(i))
Next i
' рассчитать количество элементов массива A, кратных трём
Kol = 0
For I = 0 To Ubound(A)
    If A(i) Mod 3 = 0 Then Kol = Kol + 1
Next i
' вывести рассчитанное количество в Label3
Label3.Text = "Ответ " + Str(Kol)
' очистить содержимое массива A
Erase A
End Sub
    
```

4. Сохранить проект.
5. Проверить работу проекта.

Задача 2. Создать приложение, определяющее максимальный элемент массива $D [n]$, n – количество элементов массива. Использовать `InputBox` для задания размера и значений элементов массива. Вывести элементы массива в `Label`.

1. Создать папку `Lab52`.
2. Добавить на форму элементы (рис. 13) и задать свойства (табл. 11).

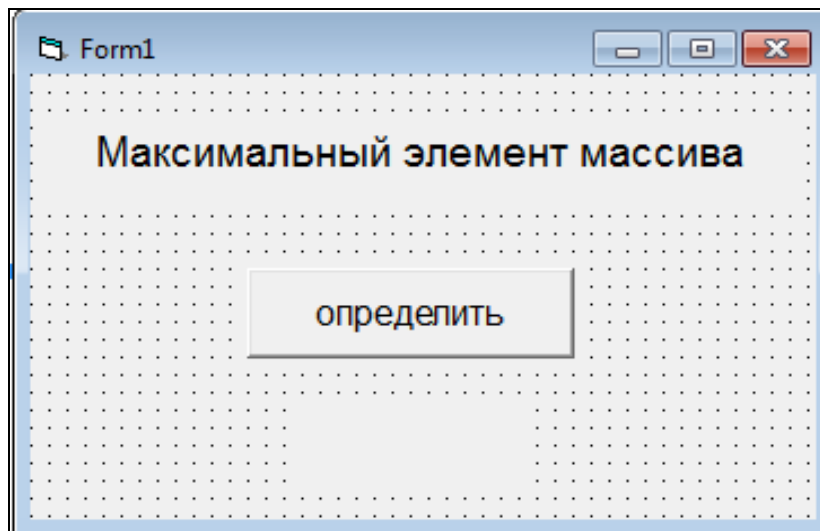


Рис. 13. Вид формы Максимальный элемент массива

Свойства элементов

Элемент	Свойство	Значение
Label1	Text	Максимальный элемент массива
Label2	Text	
Label3	Text	
Button1	Text	Определить

3. Для элемента **Button1** написать программный код:

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim D(), n, el, i, max As Integer
    ' очистить от данных Label2
    Label2.Text = " "
    ' задать количество элементов в массиве
    n = Val(TextBox("Введите число элементов массива"))
    ' количество элементов массива D будет равно введенному n
    ReDim D(n)
    ' заполнить массив D через элемент InputBox
    For i = 0 To n
        el = Val(TextBox("Введите " & i & " элемент массива D"))
        D(i) = el
    Next i
    ' вывести элементы массива D в Label2
    For i = 0 To UBound(D)
        Label2.Text = Label2.Text & " " & CStr(D(i))
    Next i
    ' определить максимальный элемент
    max = D(1)
    For i = 0 To UBound(D)
        If D(i) > max Then max = D(i)
    Next i
    ' вывести максимум в Label3
    Label3.Text = "Ответ " + Str(max)
    ' очистить содержимое массива D
    Erase D
End Sub

```

4. Сохранить проект.
5. Проверить работу проекта.

Дополнительные задания

1. Написать программный код вычисления количества элементов массива $K(9)$ с четным порядковым номером.

2. Написать программный код, рассчитывающий сумму нечетных отрицательных элементов массива F(12).
3. Написать программный код, рассчитывающий количество элементов массива V(15), равных заданному в TextBox1.
4. Написать программный код, рассчитывающий сумму элементов массива H(17), больше среднего.
5. Написать программный код, определяющий отношение максимального элемента к минимальному элементу массива G(7).

Лабораторная работа № 6

Тема: двумерные массивы

Массивы, состоящие из нескольких строк и столбцов, называются **двумерными**.

A (n,m)

n – номер строки
m – номер столбца

Например, объявить целочисленный массив A из 4 строк и 5 столбцов:
Dim A (3, 4) as Integer

Элементы этого массива имеют индексы, перечисленные в табл. 12.

Таблица 12

Элементы двумерного массива A (3, 4)

A(0, 0)	A(0, 1)	A(0, 2)	A(0, 3)	A(0, 4)
A(1, 0)	A(1, 1)	A(1, 2)	A(1, 3)	A(1, 4)
A(2, 0)	A(2, 1)	A(2, 2)	A(2, 3)	A(2, 4)
A(3, 0)	A(3, 1)	A(3, 2)	A(3, 3)	A(3, 4)

Чтобы ввести в оперативную память элементы массива A (3, 4), используя InputBox, и вывести этот массив в текстовое окно, используется программный код:

```
Dim A(3, 4) As Integer
```

```
Dim i, j As Integer
```

' заполнение числовыми значениями элементов массива

```
For i = 0 To 3
```

```
    For j = 1 To 4
```

```
        A(i, j) = Val(InputBox("Введите элемент " & i & " строки " & j & " столбца"))
```

```
    Next j
```

```
Next i
```

' вывод элементов массива в Label

```
For i = 0 To 3
```

```
    For j = 1 To 4
```

```
        Label1.Text = Label1.Text + Str(A(i, j)) & " "
```

```
Next j
Label1.Text= Label1.Text & vbCrLf
Next i
```

Задание

Задача 1. Создать приложение, выводящее в Label элементы двумерного массива **A**, размерность массива и значения элементов произвольные.

1. Создать папку Lab61.
2. Добавить на форму элементы (рис. 14) и задать свойства (табл. 13).

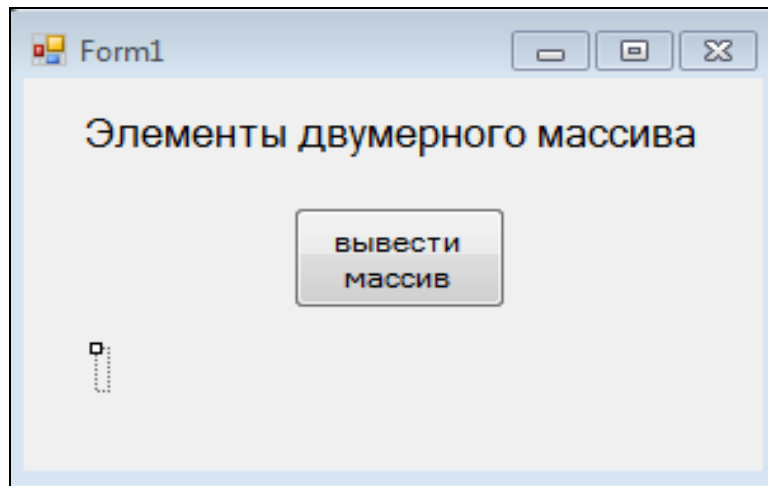


Рис. 14. Вид формы Элементы двумерного массива

Таблица 13

Свойства элементов

Элемент	Свойство	Значение
Label1	Text	Элементы двумерного массива
Label2	Text	
Button1	Text	Вывести массив

3. Для элемента **Button1** написать программный код:


```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim A(,) As Integer
    Dim i, j, n, m As Integer
    ' задать количество строк (N) и столбцов (m) массива через InputBox
    n = Val(InputBox("Введите количество строк массива A"))
    m = Val(InputBox("Введите количество столбцов массива A"))
    ' установить размер массива как n-строк, m-столбцов
    ReDim A(n, m)
```

```

' заполнить массив элементами
For i = 0 To n
    For j = 0 To m
        A(i, j) = Val(InputBox("Введите элемент " & i & " строки " & j & "
столбца"))
    Next j
Next i
' вывести двумерный массив в текстовое поле
For i = 0 To n
    For j = 0 To m
        Label2.Text = Label2.Text & Str(A(i, j)) & " "
    Next j
    Label2.Text = Label2.Text & vbCrLf
Next i
End Sub

```

4. Сохранить проект.
5. Проверить работу проекта.

Задача 2. Создать приложение, выводящий номер строки и столбца отрицательных элементов двумерного массива Q произвольного размера. Ответ вывести в Label.

1. Создать папку Lab62.
2. На форму добавить элементы Label1 (для заголовка), Label2 (для вывода массива Q), Label3 (для вывода номеров столбцов, содержащих отрицательные элементы), Button1.

3. Для элемента Button1 написать программный код:

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim Q(,) As Integer
    Dim i, j, n, m As Integer

```

' задать количество строк (n) и столбцов (m) через InputBox

```

n = Val(InputBox("Введите количество строк массива Q"))
m = Val(InputBox("Введите количество столбцов массива Q"))

```

' установить размер массива как n-строк, m-столбцов

```

ReDim Q(n, m)

```

' заполнить массив Q элементами

```

For i = 0 To n
    For j = 0 To m
        Q(i, j) = Val(InputBox("Введите элемент " & i & " строки " & j & " столбца"))
    Next j
Next i

```

' вывести двумерный массив в элемент Label2

```

For i = 0 To n
    For j = 0 To m
        Label2.Text = Label2.Text & Str(Q(i, j)) & " "
    Next j
    Label2.Text = Label2.Text & vbCrLf
Next i

```

' проверить условие отрицательности числа, истинные значения добавляем в элемент Label3

```
For i = 0 To n
    For j = 0 To m
        If Q(i, j) < 0 Then Label3.Text = Label3.Text + Str(i) + Str(j) + ","
    Next j
Next i
End Sub
```

4. Проверить работу приложения.

5. Сохранить приложение.

Задача 3. Создать приложение, рассчитывающее сумму по строке элементов массива $F(4,2)$. Ответ вывести в Label.

1. Создать папку Lab63.

2. Добавить на форму элементы Label1 (для заголовка), Label2 (для вывода массива F), Label3 (для вывода суммы по строкам), Button1.

3. Для элемента Button1 написать программный код:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
    Dim F(4, 2), sum(4) As Integer
```

```
    Dim i, j As Integer
```

' заполнить элементы массива значениями

```
For i = 0 To 4
    For j = 0 To 2
        F(i, j) = Val(InputBox("Введите элемент " & i & " строки" & j & " столбца"))
    Next j
Next i
```

' вывести двумерный массив в элемент Label2

```
For i = 0 To 4
    For j = 0 To 2
        Label2.Text = Label2.Text & Str(F(i, j)) & " "
    Next j
    Label2.Text = Label2.Text & vbCrLf
Next i
```

' обнулить расчетную переменную суммы Sum

```
sum(4) = 0
```

' рассчитать сумму по строке

```
For i = 0 To 4
    For j = 0 To 2
        sum(i) = sum(i) + F(i, j)
    Next j
```

' добавить рассчитанную по строке массива сумму в Label3

```
Label3.Text = Label3.Text + Str(sum(i)) & " "
```

' обнулить переменную Sum для расчета суммы элементов следующей строки

```
sum(i) = 0
```

```
Next i
```

```
End Sub
```

4. Сохранить приложение.
5. Проверить работу приложения.

Дополнительные задания

1. Написать программный код, рассчитывающий количество нечетных элементов по столбцу массива $F(n,m)$.
2. Написать программный код, выводящий номер строки и столбца нулевых элементов массива $V(n,m)$.
3. Написать программный код, рассчитывающий количество отрицательных элементов по столбцу двумерного массива $W(2, 5)$.
4. Написать программный код, рассчитывающий сумму элементов с четным номером строки и нечетным номером столбца двумерного массива $Z(n, m)$.
5. Написать программный код, выводящий в Label положительные элементы главной диагонали квадратной матрицы $R(5, 5)$.

Используемая литература

Казанский А.А. Объектно-ориентированное программирование на Visual Basic 2010 и Visual C# 2010 в среде разработки Microsoft Visual Studio: учебное пособие и практикум. М-во образования и науки РФ, ФГБОУ ВПО «Моск. гос. строит. ун-т». – Москва: МГСУ, 2012. 422 с.

Содержание

Введение	3
Лабораторная работа № 1	3
Лабораторная работа № 2	10
Лабораторная работа № 3	13
Лабораторная работа № 4	17
Лабораторная работа № 5	19
Лабораторная работа № 6	24
Используемая литература	28