

*Авторская разработка преподавателя информатики и ИКТ
Курбатова Вячеслава Валерьевича, г. Гусев*

ПЕРЕМЕННЫЕ

Переменные в программировании имеют такой же смысл, как в математике. Перед тем, как использовать переменную, ее рекомендуется описать (объявить).

Синтаксис оператора описания переменной:

Dim переменная [As тип]

В этой конструкции:

Dim – ключевое слово, свидетельствующее о том, что объявляется переменная (dimension - размер);

переменная – имя объявляемой переменной;

As – ключевое слово, используемое при задании типа данных (as - как);

тип – тип данных для объявляемой переменной или (что то же самое) тип переменной.

Dim переменная [As тип]

Когда при выполнении программы компьютер встречает оператор **Dim**, он выделяет переменной **переменная** часть своей оперативной памяти, которую в программировании принято называть ячейкой. Размер выделенной ячейки, исчисляемый в байтах, определяется типом переменной **тип**.

Размеры ячеек, соответствующих разным переменным, могут сильно отличаться.

Типы переменных:

Тип переменной – значение, которое может принимать ячейка.

Integer , **Long** – целые числа (1001)

Currency , **Single** , **Double** - числа с дробной частью
(3,1415926)

Currency - Тип с плавающей запятой с 4 десятичными числами, используемыми для финансовых значений

Single - Тип с плавающей запятой, поддерживающий приблизительно 7 цифр точности

Double - Тип с плавающей запятой, поддерживающий приблизительно 15 цифр точности

Типы переменных:

String – строковая (текстовая) информация
(“Здравствуй Мир!”)

Boolean – логическое значение (True / False)

Byte – двоичное значение (натуральные числа и ноль)

Date – время или дата

Если при объявлении переменной не указан ее **тип**, то автоматически принимается тип **Variant**, который остается неопределенным до первого программного действия над данной переменной.

Если действие арифметическое, то переменная автоматически становится числовой, если же действие символьное (текстовое), то переменная становится символьной и т.д.

Пример:

Dim A01 As Integer

Dim A02 As String

Dim A03 As Boolean

Dim A04 As Byte

Пример:

Dim A01 As Integer, A02 As String, A03 As Boolean, A04 As Byte

Dim A01, B01, C01, D01 As Integer

КОНСТАНТЫ

Константа, такая же переменная, как и простая переменная, но в отличие от нее, значение объявленной константы невозможно в дальнейшем изменить.

Const константа [As тип] = значение

Пример:

Const pi As Double = 3.1415926654

Const e As Double = 2.718281828

Const Message = “Завершение работы”

Const Millenium As Date = # 1 Jan 2000 #

Const Beta As Currency = 1/3

Const Min = 0, Max = 1000, Flag As Boolean = False

ДВЕ ОСНОВНЫЕ ФУНКЦИИ ПРЕОБРАЗОВАНИЯ ТИПОВ ДАННЫХ:

VAL () и STR()

VAL () – преобразование строки в число

STR() – преобразование числа в строку

Пример:

Sub StrVal()

Dim strA As String

Dim curB As Currency

strA = "45.77"

curB = Val(strA)

‘ Результат: curB = 45.77

strA = Str(curB)

‘ Результат: strA = " 45.77" (добавляет 1 пробел спереди)

curB = Val("4.7 = x")

‘ Результат: curB = 4.7

curB = Val("x = 4.7")

‘ Результат: curB = 0

End Sub

Пример:

Dim MyValue

MyValue = Val("2457")

‘ Результат: MyValue = 2457

MyValue = Val(" 2 45 7")

‘ Результат: MyValue = 2457

MyValue = Val("24 and 57")

‘ Результат: MyValue = 24

АРИФМЕТИЧЕСКИЕ ВЫРАЖЕНИЯ

Математическое выражение	Арифметическое выражение VB
$5x + 12y$	$5 * x + 12 * y$
$\frac{x}{y}$	x / y
y^x	$y ^ x$
x	x
$19.55 \cdot 10^{-6}$	19.55E-6 или 19.55D-6

/ – деление одного числа на другое

**** – целочисленное деление одного числа на другое (аналог оператора Dim)

Mod – остаток от целочисленного деления

Пример:

Sub Арифметика1()

Dim m As Integer, n As Integer, x As Double

m = 5 : n = 2

x = m / n ‘ Результат: x = 2.5

x = m \ n ‘ Результат: x = 2

x = m Mod n ‘ Результат: x = 1

End Sub

Пример:

x = - x ‘ Результат: изменение знака числа

Математическое выражение	Арифметическое выражение VB
$-a^b$	$-a \wedge b$ или $-a \wedge (b)$
a^{-b}	$a \wedge (-b)$
a^{b+c}	$a \wedge (b+c)$
$10^{-4.7}$	$10 \wedge (-4.7)$
$10^{4.7}$	$10 \wedge 4.7$
a^{b^c}	$a \wedge (b \wedge c)$
$(a^b)^c$	$a \wedge b \wedge c$ или $(a \wedge b) \wedge c$
$\frac{a \cdot b}{c \cdot d}$	$a * b / (c * d)$ или $(a * b) / (c * d)$
$a \cdot 10^4$	$a * 1E4$, $a * 1D4$ или $a * 1000$

Пример:

Sub Арифметика2()

Dim m As Integer, n As Integer

Dim x As Single, y As Single

x = 3: m = 2: n = - 1

y = (- 3) ^ m

‘ Результат: y = 9

y = - (3 ^ m)

‘ Результат: y = -9

y = - 3 ^ m

‘ Результат: y = 9

y = 10 + (x + 7) ^ (m + n)

‘ Результат: y = 20

y = 10 + x + 7 ^ m + n

‘ Результат: y = 61

End Sub

ЛОГИЧЕСКИЕ ВЫРАЖЕНИЯ

Помимо арифметических выражений, в VB можно использовать логические выражения (утверждения), принимающие одно из двух значений типа **Boolean** – **True** (истина, логическая единица) или **False** (ложь, логический нуль).

5 >= 3

‘ Результат True

5 < 3

‘ Результат False

Пример:

Sub Логика1()

Dim x As Integer

Dim y As Integer

Dim blnA As Boolean

x = 5

y = 2

blnA = x > y

‘ Результат blnA = True

blnA = x = y

‘ Результат blnA = False

End Sub

Более сложные **логические выражения** составляются с помощью логических операций.

Рассмотрим три из них – **Not**, **And** и **Or**.

Операция **Not** определяется следующим образом:

- если A равно True, то Not A равно False;
- если A равно False, то Not A равно True.

Определение операции And

A	B	A And B
True	True	True
True	False	False
False	True	False
False	False	False

Определение операции Or

A	B	A Or B
True	True	True
True	False	True
False	True	True
False	False	False

При наличии в выражении нескольких логических операций порядок их выполнения определяется следующим правилом **приоритетов**:

- 1) в **первую** очередь выполняется операция **Not**;
- 2) **далее** выполняется операция **And**;
- 3) в **последнюю** очередь выполняется операция **Or**.

Для изменения последовательности логических операций используются круглые скобки (как в арифметических выражениях).

Пример:

Sub Логика1()

Dim x, y, z As Double

Dim blnA As Boolean

X = 1 : Y = 2.87 : Z = 3.12

blnA = (x > y) And (y < z)

‘ Результат blnA = False

blnA = x < y And y < z

‘ Результат blnA = True

blnA = x > y Or y > z

‘ Результат blnA = False

blnA = Not (x < y Or Not y < z)

‘ Результат blnA = False

blnA = Not x > y And x > y

‘ Результат blnA = False

blnA = Not (x > y And x > y)

‘ Результат blnA = True

End Sub

Оператор перехода

GoTo метка

Метка – целое неотрицательное число без знака (0, 1, 2, 3, ...) или последовательность букв и цифр, начинающаяся с буквы (например, start53a)

Sub Programma2()

Dim X As Integer

X = 12

➡ 2: If $X > 9$ And $X < 12$ Then GoTo LastLine

$$X = X - 2$$

GoTo 2

LastLine: 

End Sub

Конструкция принятия решений

If . . . Then . . .

Конструкция **If . . . Then** называется условным оператором.

Простейший условный оператор:

If условие **Then** оператор

```
Sub Programma2()  
    Dim X As Integer  
    X = 12                ' начальное значение X  
    If (X > 9 And X < 12) Then X = X + 1  
    X = X + 2             ' конечное значение X  
    X = X * 2  
End Sub
```

If условие Then GoTo метка

Переход по условию

```
Sub Programma2()
```

```
    Dim X As Integer
```

```
    X = 12
```

```
2:   If X > 9 And X < 12 Then GoTo LastLine
```

```
        X = X - 2
```

```
        GoTo 2
```

```
LastLine: ←
```

```
End Sub
```



If условие **Then**

операторы 1

Else

операторы 2

End If

Если условие выполняется, то

выполняются операторы 1, иначе

выполняются операторы 2

Sub Vybor()

Dim X As Integer

X = 12

If (X > 9 And X < 12) Then

X = X + 3

Else

X = X + 2

End If

End Sub

Циклы

For счетчик = начало **To** конец [**Step** шаг]

Операторы

Next [счетчик]

Цикл **For . . . Next** используется в том случае, когда количество выполнений заданного блока операторов известно заранее.

Sub Рассчет1()

Dim I As Integer

Dim F As Integer

F = 1

For I = 1 To 6 Step 2

F = F * I

Next I

For I = 1 To F ^ 2

G = G + I

Next I

End Sub

Цикл **While . . . Wend** и **Do While . . . Loop** применяется, когда число выполнений операторов цикла заранее неизвестно. Цикл выполняется до тех пор, пока **условие** истинно.

Do While **условие**
операторы
Loop

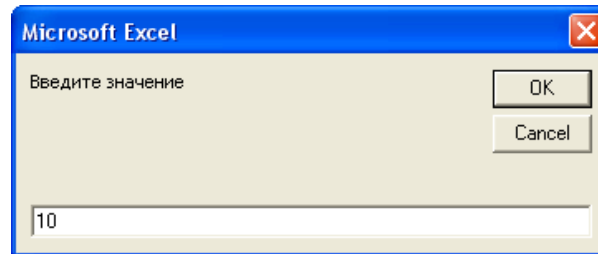
While **условие**
операторы
Wend

Do
операторы
Loop Until **условие**

По оператору **Exit Do** можно выйти из цикла по условию.

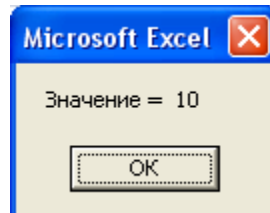
Использование стандартных окон операционной системы Windows

Ввод в переменную `Peremennaya` значения с клавиатуры



`Peremennaya = InputBox(«пояснительный текст»)`

Вывод значения `Peremennaya` на экран

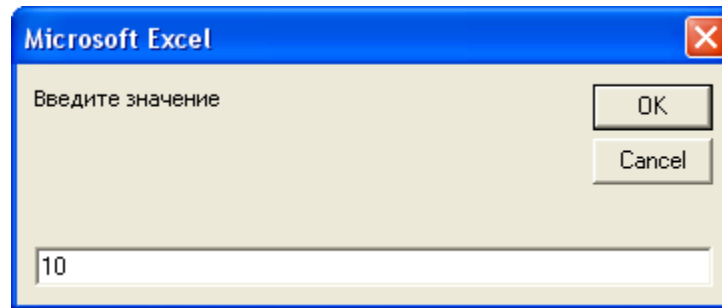


`MsgBox «Значение = », Peremennaya`

Sub Vvod_Vyvod()

Dim Peremennaya As Integer

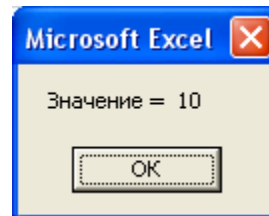
Peremennaya = InputBox("Введите значение")



a = "Значение = " & Str(Peremennaya)

MsgBox a

End Sub



Первая программа



Главная

Вставка

Разметка страницы

Формулы

Данные

Рецензирование

Вид

Разработчик

Visual
Basic

Макросы



Запись макроса



Относительные ссылки



Безопасность макросов

Код



Вставить

Режим
конструктора

Элементы управления



Свойства



Просмотр кода



Отобразить окно



Источник



Свойства кар...



Пакеты расширения



Обновить данные

XML



Импорт



Экспорт

Область
документа
Изменить

R40C20

fx

1

2

3

4

5

6

7

8

9

10

11

12

13

14

1

2

3

4

5

6

7

8

9

10

11

12

Открываем вкладку
«Разработчик»



Главная

Вставка

Разметка страницы

Формулы

Данные

Рецензирование

Вид

Разработчик

Visual
Basic

Макросы



Запись макроса



Относительные ссылки



Безопасность макросов

Код



Вставить

Режим
конструктора

Свойства



Просмотр кода



Отобразить окно



Источник



Свойства карты



Пакеты расширения



Обновить данные

XML



Импорт



Экспорт

Область
документа
Изменить

R40C20

fx

1

2

3

4

8

9

10

11

12

13

14

1

2

3

4

5

6

7

8

9

10

11

12

Элементы управления формы



Элементы ActiveX



Кнопка (элемент ActiveX)

Вставка элемента управления "кнопка".

Выбираем элемент
«Кнопка»



Главная

Вставка

Разметка страницы

Формулы

Данные

Рецензирование

Вид

Разработчик



Visual Basic



Макросы

Запись макроса

Относительные ссылки

Безопасность макросов

Код



Вставить



Режим конструктора

Элементы управления



Свойства

Просмотр кода

Отобразить окно



Источник



Свойства карты

Пакеты расширения

Обновить данные

XML



Импорт



Экспорт



Область документа

Изменить

CommandButton1 fx =ВНЕДРИТЬ("Forms.CommandButton.1";")

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														

CommandButton1

Отредактируем
надпись на кнопке
Нажимаем на
кнопке ПКМ



Главная

Вставка

Разметка страницы

Формулы

Данные

Рецензирование

Вид

Разработчик

Visual
Basic

Макросы



Запись макроса



Относительные ссылки



Безопасность макросов

Код



Вставить

Режим
конструктора

Элементы управления



Свойства



Просмотр кода



Отобразить окно



Источник



Свойства карты



Пакеты расширения



Обновить данные

XML



Импорт



Экспорт

Область
документа
Изменить

CommandButton1 fx =ВНЕДРИТЬ("Forms.CommandButton.1";")

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														

Выбираем Edit



Главная

Вставка

Разметка страницы

Формулы

Данные

Рецензирование

Вид

Разработчик

Visual
Basic

Макросы



Запись макроса



Относительные ссылки



Безопасность макросов

Код



Вставить

Режим
конструктора

Элементы управления



Свойства



Просмотр кода



Отобразить окно



Источник



Свойства карты



Пакеты расширения



Обновить данные

XML



Импорт



Экспорт

Область
документа

Изменить

R41C20

1

2

**В свойствах
изменяем
внутреннее имя
объекта «Кнопка»**

Name = Vvod_Vyvod

8

9

10

11

12

13

14

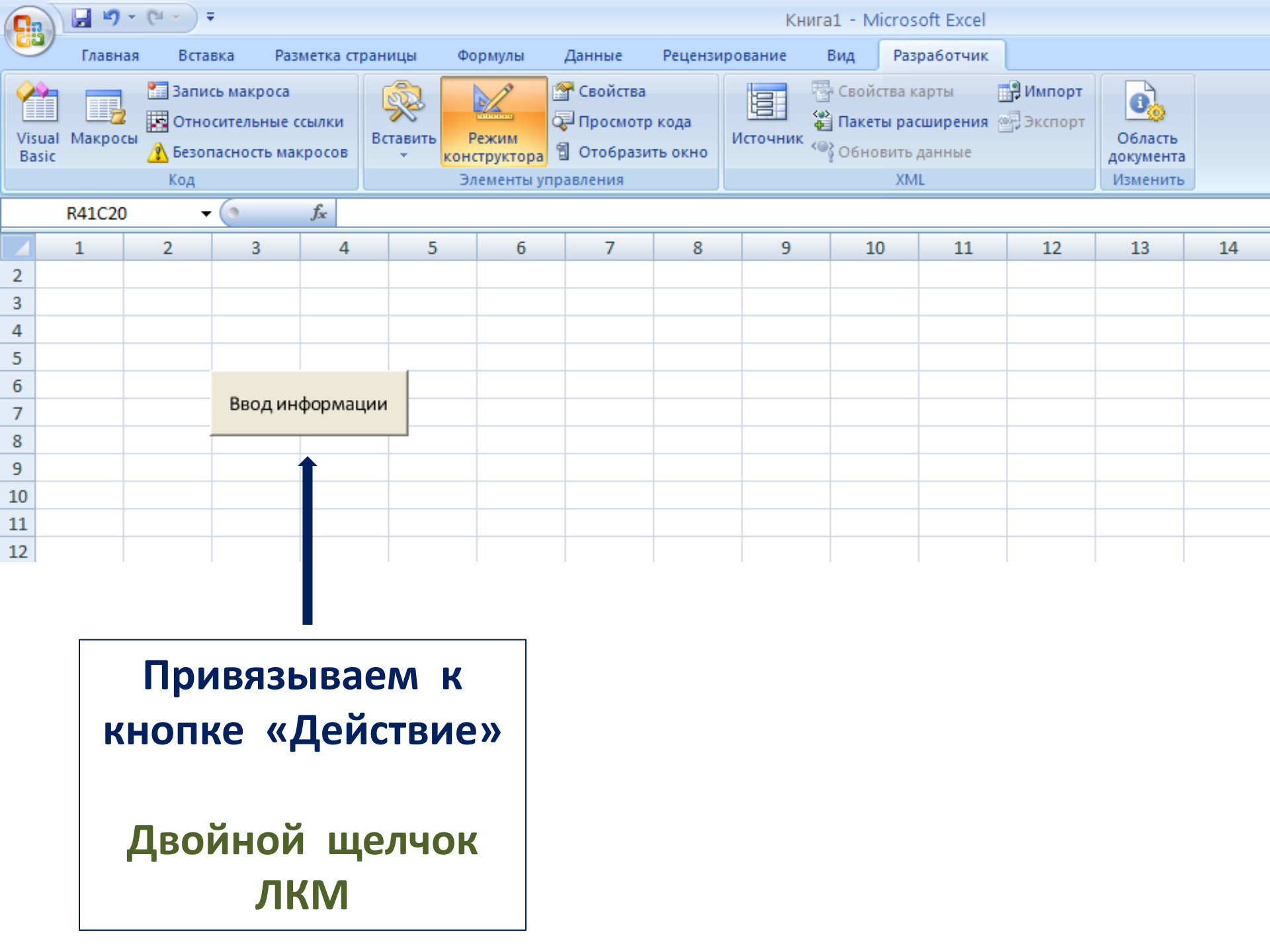
Properties

Vvod_Vyvod CommandButton

Alphabetic Categorized

(Name) Vvod_Vyvod

Accelerator	
AutoLoad	False
AutoSize	False
BackColor	&H8000000F&
BackStyle	1 - fmBackStyleOpaque
Caption	CommandButton1
Enabled	True
Font	Calibri
ForeColor	&H80000012&
Height	24
Left	336,75
Locked	True
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Picture	(None)
PicturePosition	7 - fmPicturePositionAboveCenter
Placement	2
PrintObject	True
Shadow	False
TakeFocusOnClick	True
Top	37,5
Visible	True
Width	72
WordWrap	False



Visual Basic Макросы

Запись макроса
Относительные ссылки
Безопасность макросов
Код

Вставить

Режим конструктора
Элементы управления

Свойства
Просмотр кода
Отобразить окно

Источник

Свойства карты
Пакеты расширения
Обновить данные
XML

Импорт
Экспорт

Область документа
Изменить

R41C20

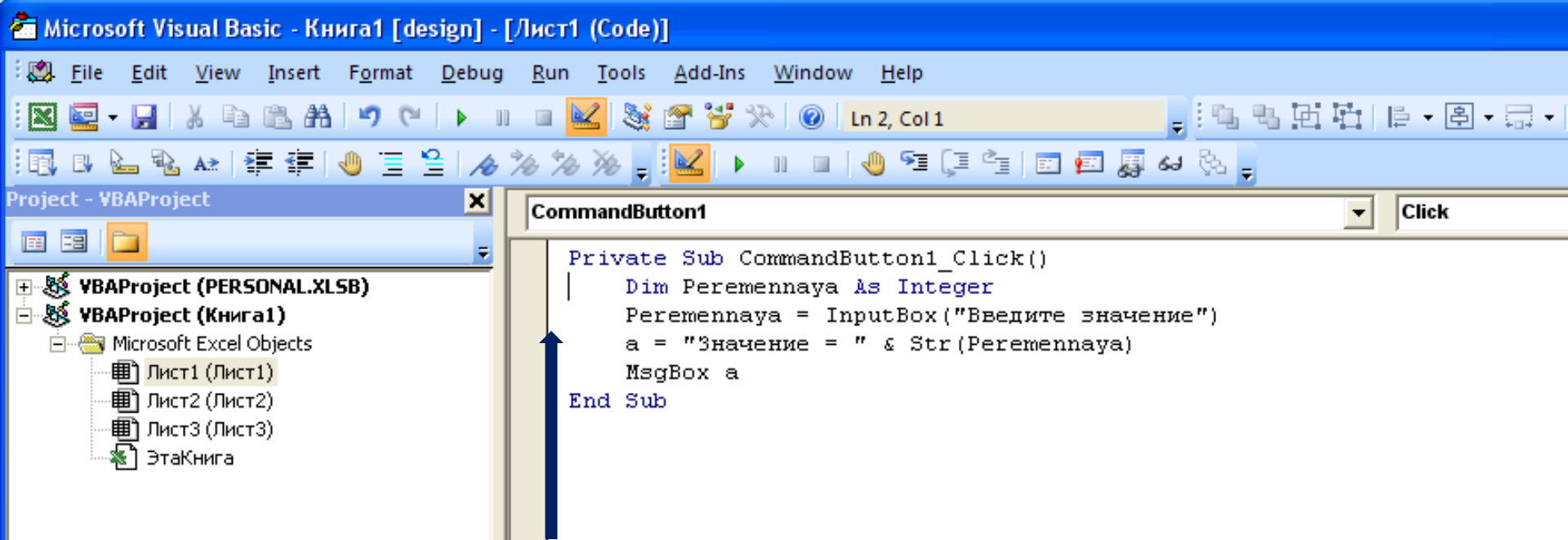
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														

Ввод информации



Привязываем к
кнопке «Действие»

Двойной щелчок
ЛКМ



Вводим программный код:

Private Sub Vvod_Vyvod()

Dim Peremennaya As Integer

Peremennaya = InputBox("Введите значение")

a = "Значение = " & Str(Peremennaya)

MsgBox a

End Sub

Выполнение программы



Главная

Вставка

Разметка страницы

Формулы

Данные

Рецензирование

Вид

Разработчик

Visual
Basic

Макросы

Запись макроса

Относительные ссылки

Безопасность макросов

Код



Вставить

Режим
конструктора

Элементы управления



Свойства



Просмотр кода



Отобразить окно



Источник



Свойства карты



Пакеты расширения



Обновить данные

XML



Импорт



Экспорт

Область
документа

Изменить

R41C20

fx

1 2 3 4 5 6 7 8 9 10 11 12 13 14

2

3

4

5

6

7

8

9

10

11

12

Ввод информации

Отключаем режим
конструктора



Главная

Вставка

Разметка страницы

Формулы

Данные

Рецензирование

Вид

Разработчик

Visual
Basic

Макросы

Запись макроса

Относительные ссылки

Безопасность макросов

Код



Вставить

Режим
конструктора

Элементы управления



Свойства

Просмотр кода

Отобразить окно



Источник



Свойства карты

Пакеты расширения

Обновить данные

XML



Импорт

Экспорт

Область
документа

Изменить

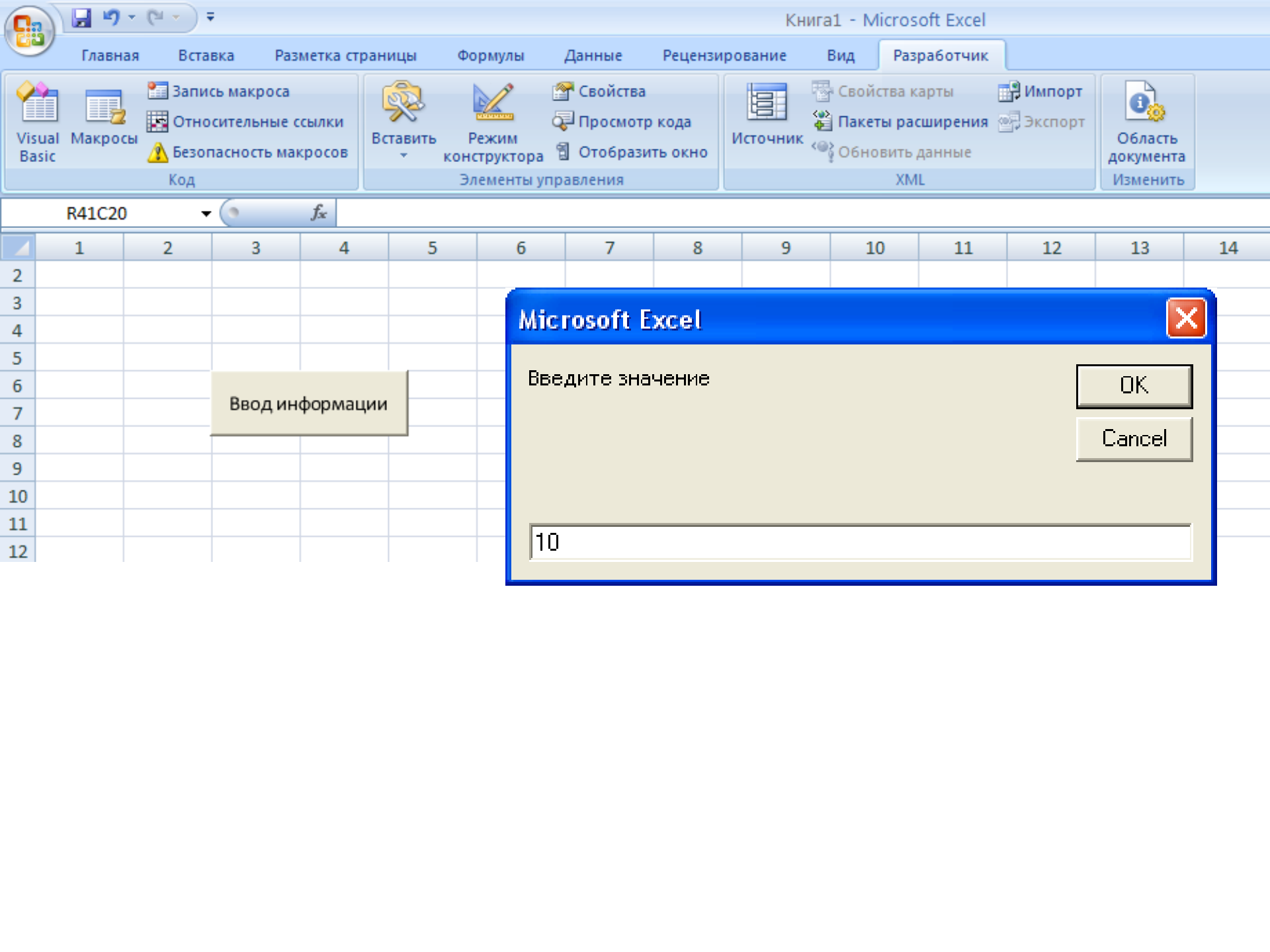
R41C20

fx

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														

Ввод информации

**Нажимаем на кнопку**



Visual Basic

Макросы

Запись макроса

Относительные ссылки

Безопасность макросов

Код

Вставить

Режим конструктора

Элементы управления

Свойства

Просмотр кода

Отобразить окно

Источники

Свойства карты

Пакеты расширения

Обновить данные

Импорт

Экспорт

XML

Область документа

Изменить

R41C20

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														

Ввод информации

Microsoft Excel

Введите значение

10

OK

Cancel



Главная

Вставка

Разметка страницы

Формулы

Данные

Рецензирование

Вид

Разработчик

Visual
Basic

Макросы



Запись макроса



Относительные ссылки



Безопасность макросов

Код



Вставить

Режим
конструктора

Свойства



Просмотр кода



Отобразить окно

Элементы управления



Источник



Свойства карты



Пакеты расширения



Обновить данные

XML



Импорт



Экспорт

Область
документа

Изменить

R41C20

fx

1

2

3

4

5

6

7

8

9

10

11

12

13

14

Microsoft Excel



Значение = 10

OK

Ввод информации

Работа с ячейками таблицы Excel

**Worksheets("ИмяЛиста").Cells(НомерСтроки,
НомерСтолбца) = ИмяПеременной**

**Занесение в ячейку таблицы значение
переменной ИмяПеременной**

**ИмяПеременной = Worksheets("ИмяЛиста").
Cells(НомерСтроки, НомерСтолбца)**

**Занесение в переменную ИмяПеременной
значение ячейки таблицы**

```
Private Sub CommandButton2_Click()
```

```
    Dim k, s As Integer
```

```
    s = 0: k = 1
```

```
        Do While k < 11
```

```
            s = s + k
```

```
            k = k + 1
```

```
            Worksheets("Лист1").Cells(k, 1) = s
```

```
        Loop
```

```
        MsgBox s
```

```
End Sub
```

Выполнение программы



Главная

Вставка

Разметка страницы

Формулы

Данные

Рецензирование

Вид

Разработчик

Visual
Basic

Макросы

Запись макроса

Относительные ссылки

Безопасность макросов

Код



Вставить

Режим
конструктора

Элементы управления



Свойства

Просмотр кода

Отобразить окно



Источник



Свойства карты

Пакеты расширения

Обновить данные

XML



Импорт

Экспорт

Область
документа

Изменить

R41C20

fx

1

2

3

4

5

6

7

8

9

10

11

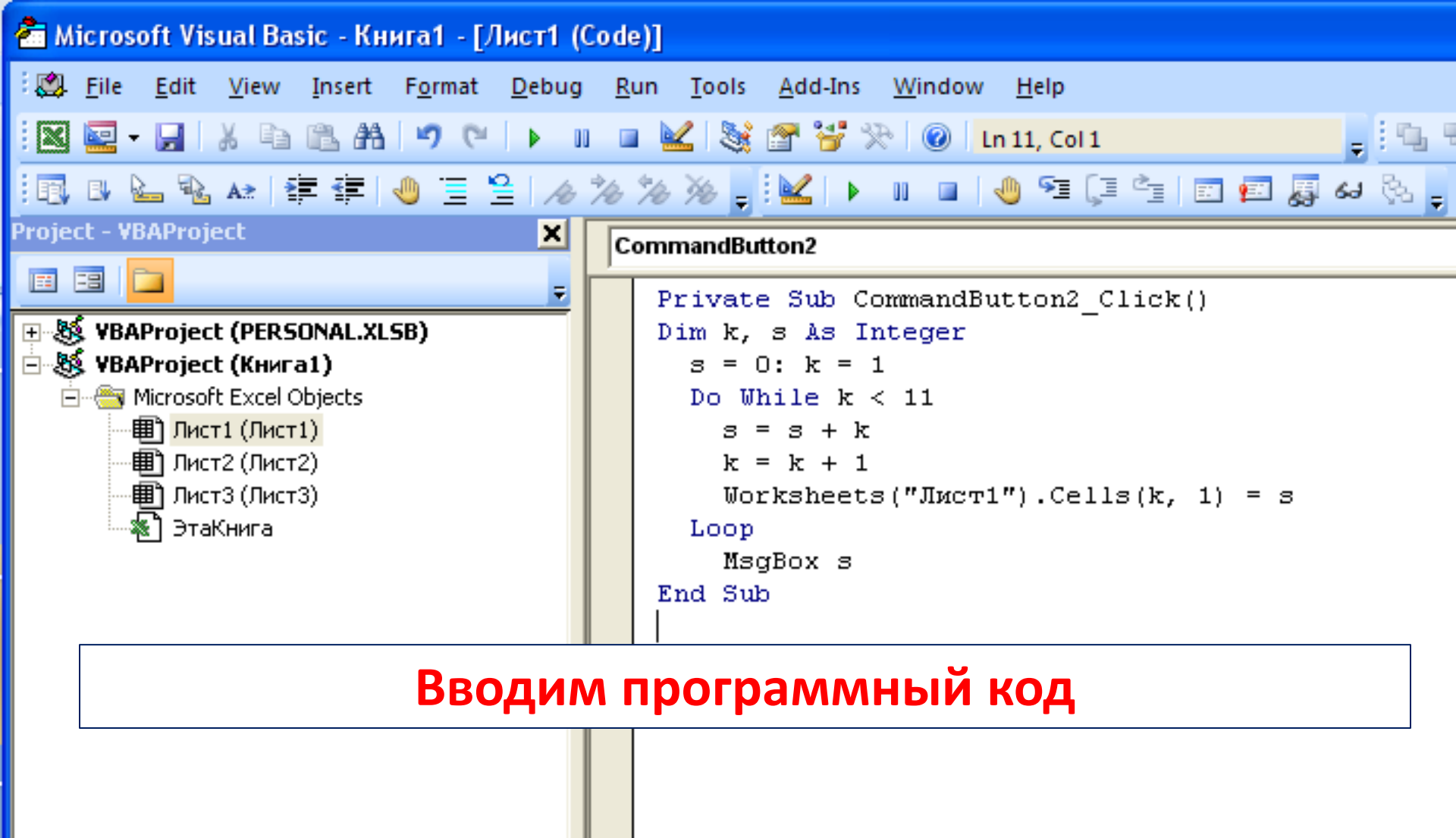
12

13

14

Расчет

Создаем кнопку
«Расчет»



Вводим программный код

Выполнение программы