

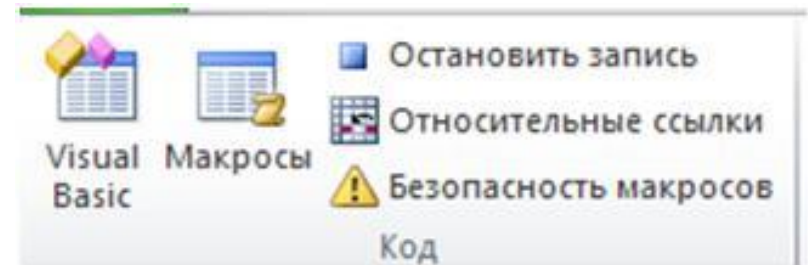
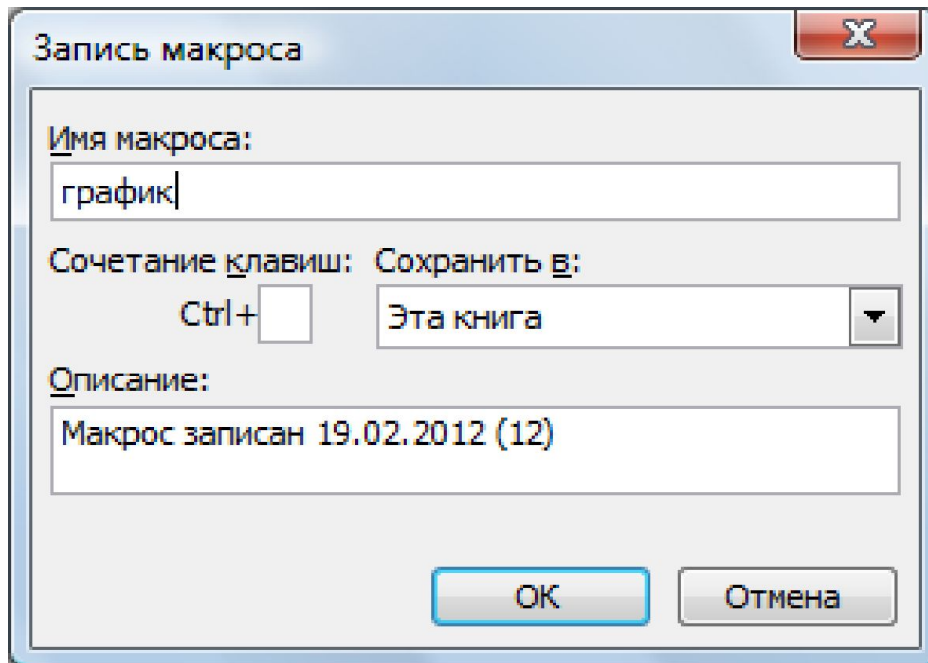
# VBA

## Visual Basic for Application (лекции 1-3)

**МАКРОС** — это именованная последовательность заданных пользователем команд и действий, хранящаяся в форме программы на языке VBA.

*MacroRecorder* — транслятор, создающий программу (макрос) на языке VBA. Для записи макроса с помощью *Макрорекодера* необходимо:

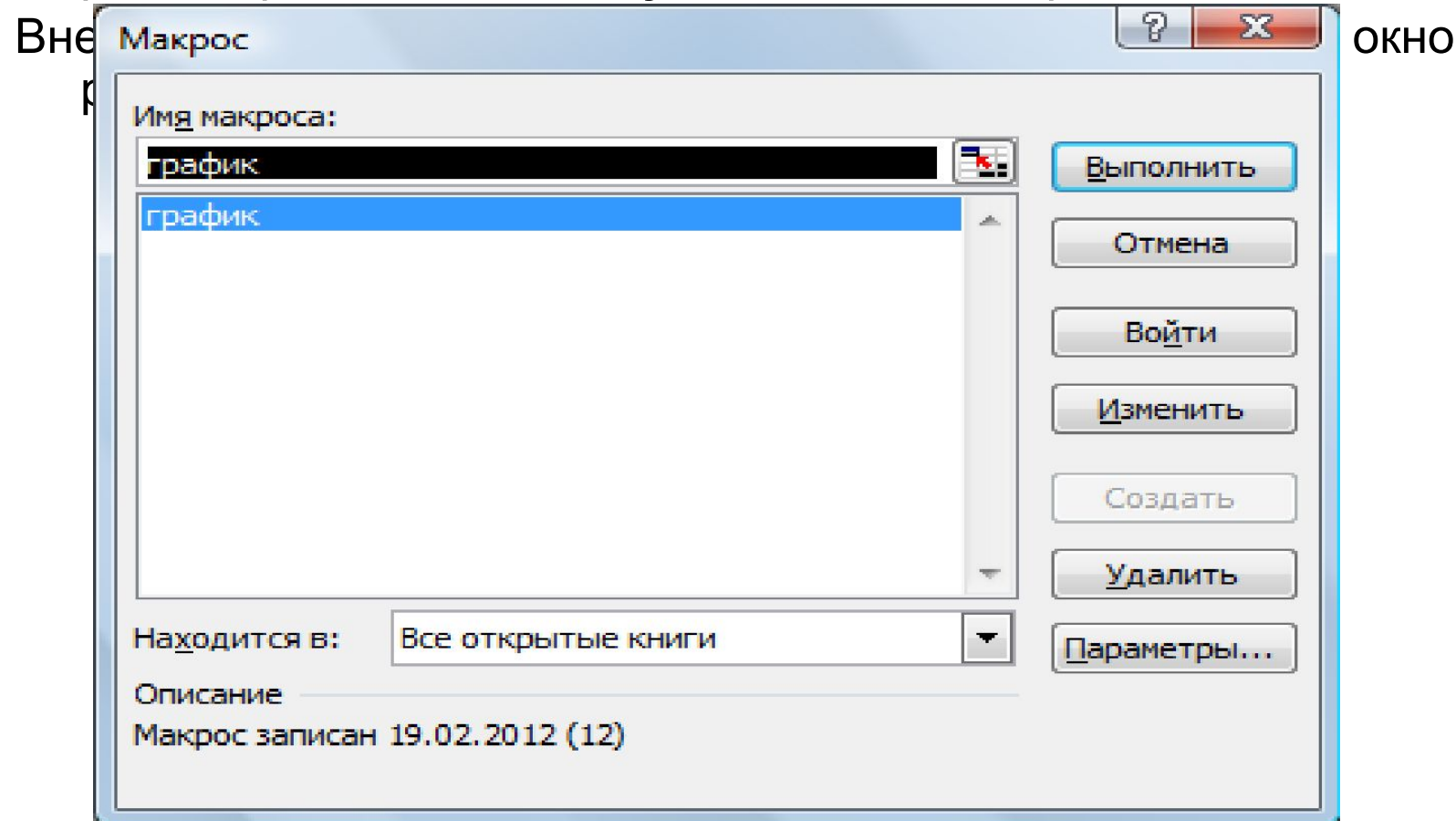
- 1. Вкладка **Разработчик**, группа **Код**, кнопка **Запись макроса** .
- 2. В окне **Запись макроса** установить параметры записываемой процедуры , нажать **ОК**. Кнопка **Запись макроса** в группе **КОД** изменится на кнопку **Остановить запись** .
- 3. Выполнить действия, которые нужно записать.
- 4. Нажать кнопку **Остановить запись**



Группа **КОД** при записи макроса

## **Для просмотра и редактирования созданного макроса:**

Вкладка **Разработчик**, группа **Код**, кнопка **Макросы**, в диалоговом окне **Макрос** выбрать имя макроса, нажать кнопку **Изменить**, откроется редактор VBA и окно **Модуль** с текстом макроса.



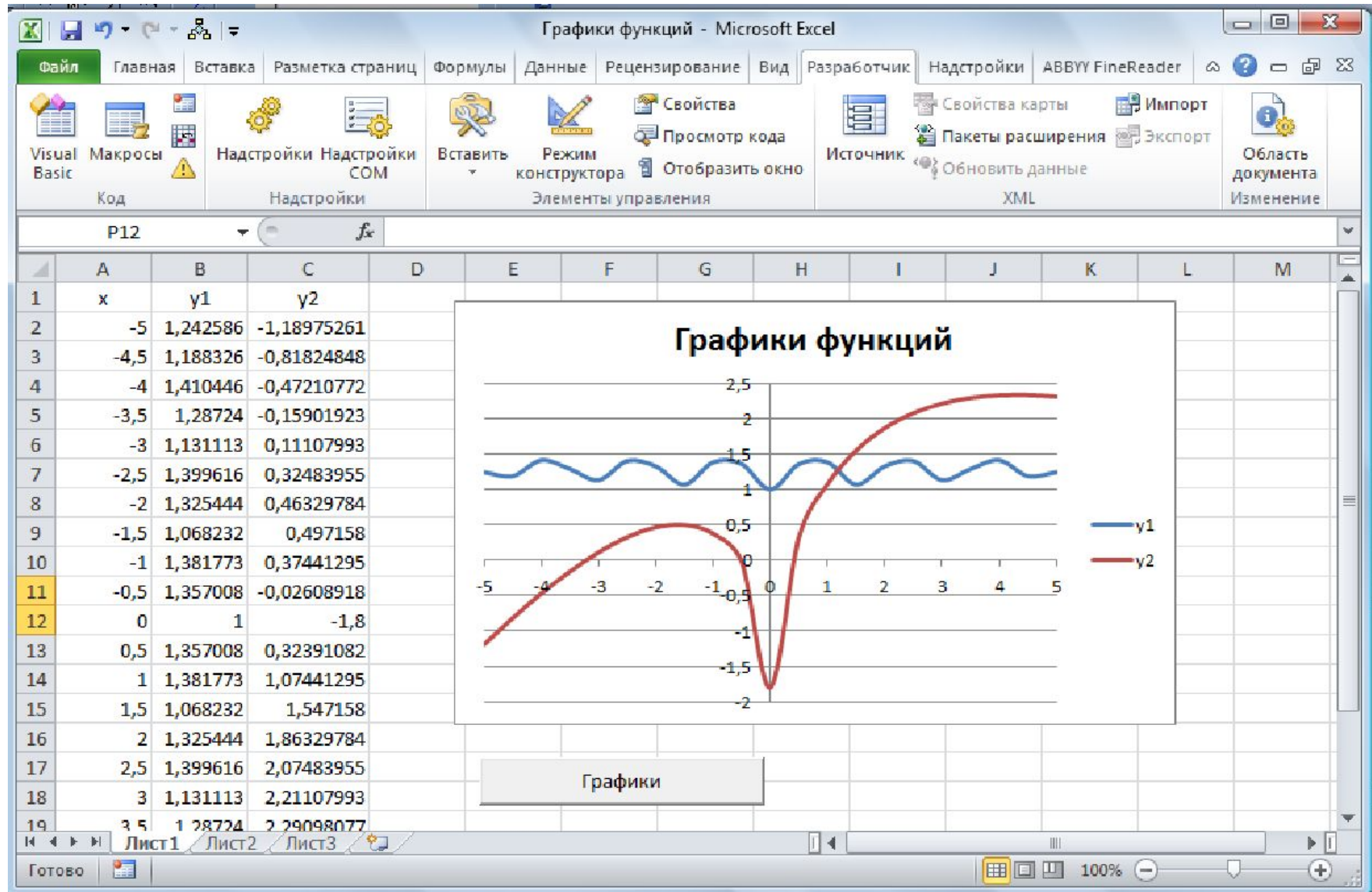
## **Для выполнения макроса:**

Вкладка **Разработчик**, группа **Код**, кнопка **Макросы**, в диалоговом окне **Макрос** выбрать имя макроса, нажать кнопку **Выполнить**

Пример: создать **макрос** построения графиков функций

$$Y_1 = |\sin x| + |\cos x| \text{ и } Y_2 = 3\sin\sqrt{x+0,35x} - 1,8$$

на отрезке  $[-5; 5]$  с шагом  $h=0,5$

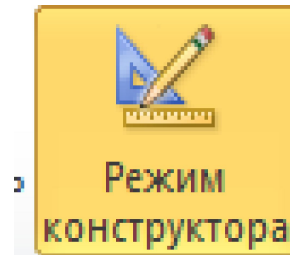


# Назначение макроса **ГРАФИК** командной кнопке

Вкладка **Разработчик**, группа **Элементы управления**, кнопка **Вставить**.

Элемент управления **Кнопка (CommandButton)** из группы **Элементы ActiveX** установить на рабочем листе.

После появления командной кнопки на рабочем листе на вкладке **Разработчик**



кнопка **Режим Конструктора** становится активной. Затем щелкнуть правой кнопкой мыши по созданной кнопке и в открывшемся контекстном меню выбрать команду «Свойства».

Откроется окно «Properties», где изменить надпись на кнопке: «Графики».

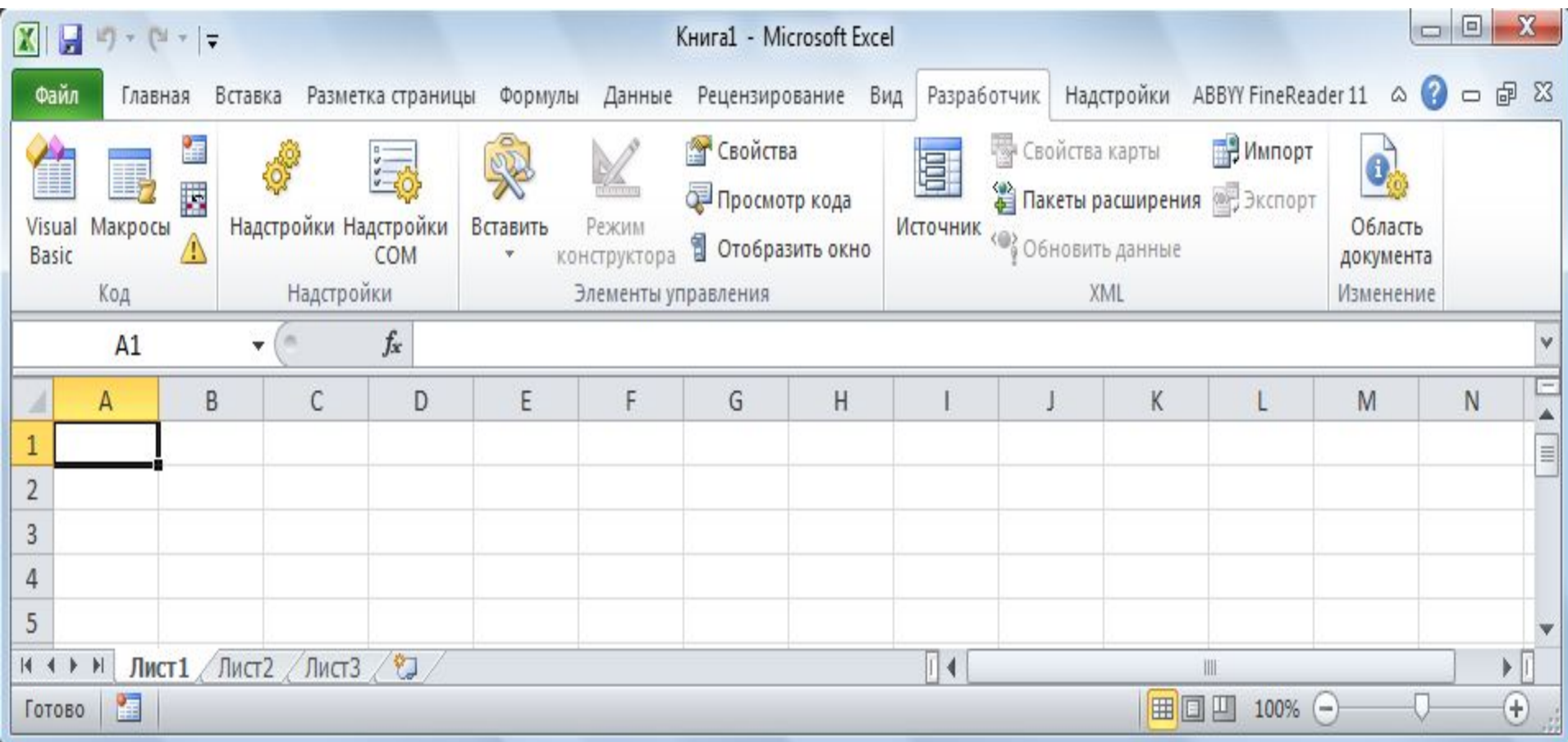
Еще раз щелкнуть правой кнопкой мыши на кнопке и в открывшемся контекстном меню выбрать команду «Исходный текст».

Откроется окно кода рабочего листа, где после заголовка процедуры, ввести инструкцию: Call график.

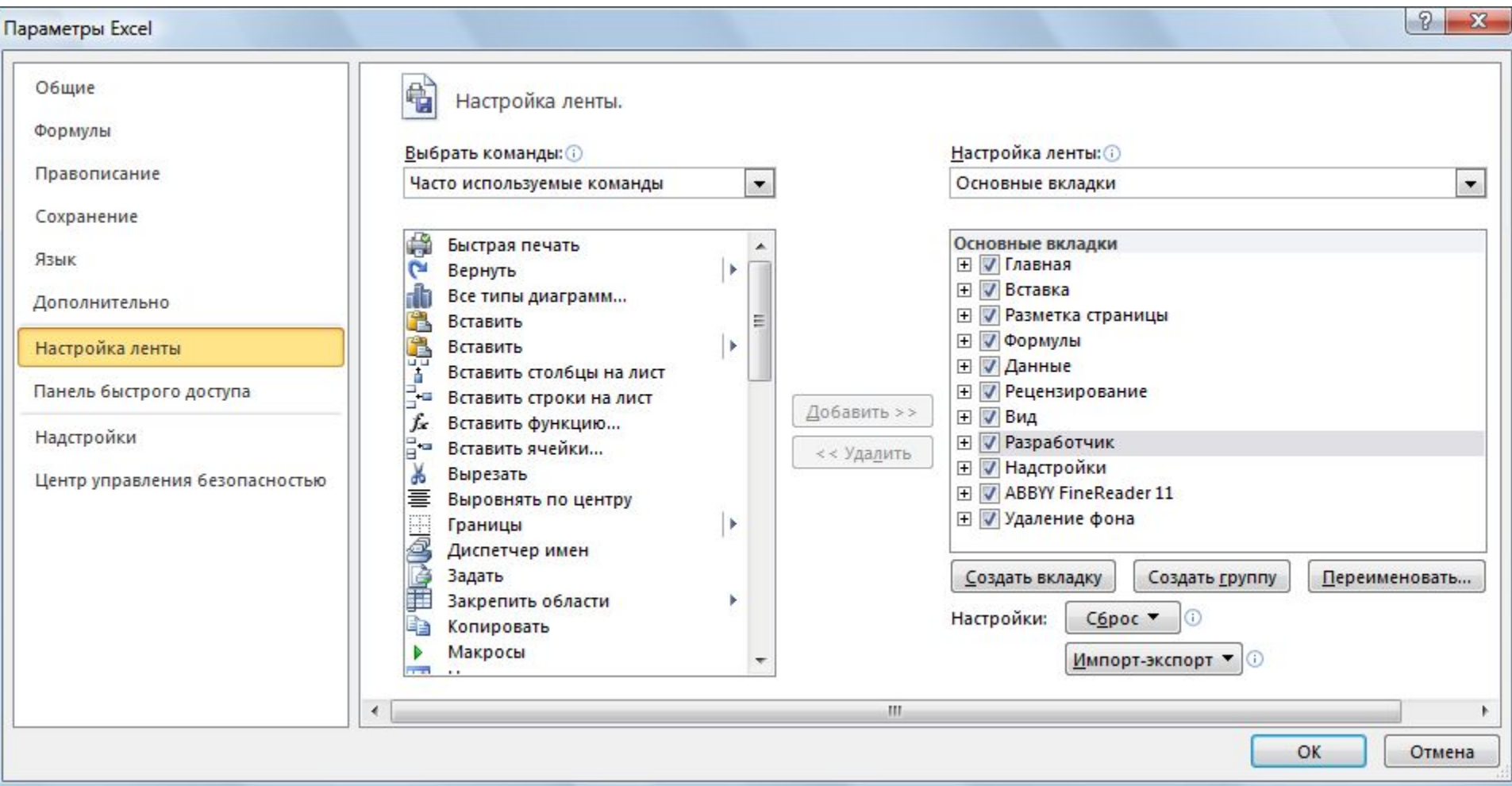
```
Private Sub CommandButton1_Click()  
Call график  
End Sub
```

Чтобы запустить макрос достаточно будет нажать кнопку «Графики».

# Редактор VBA

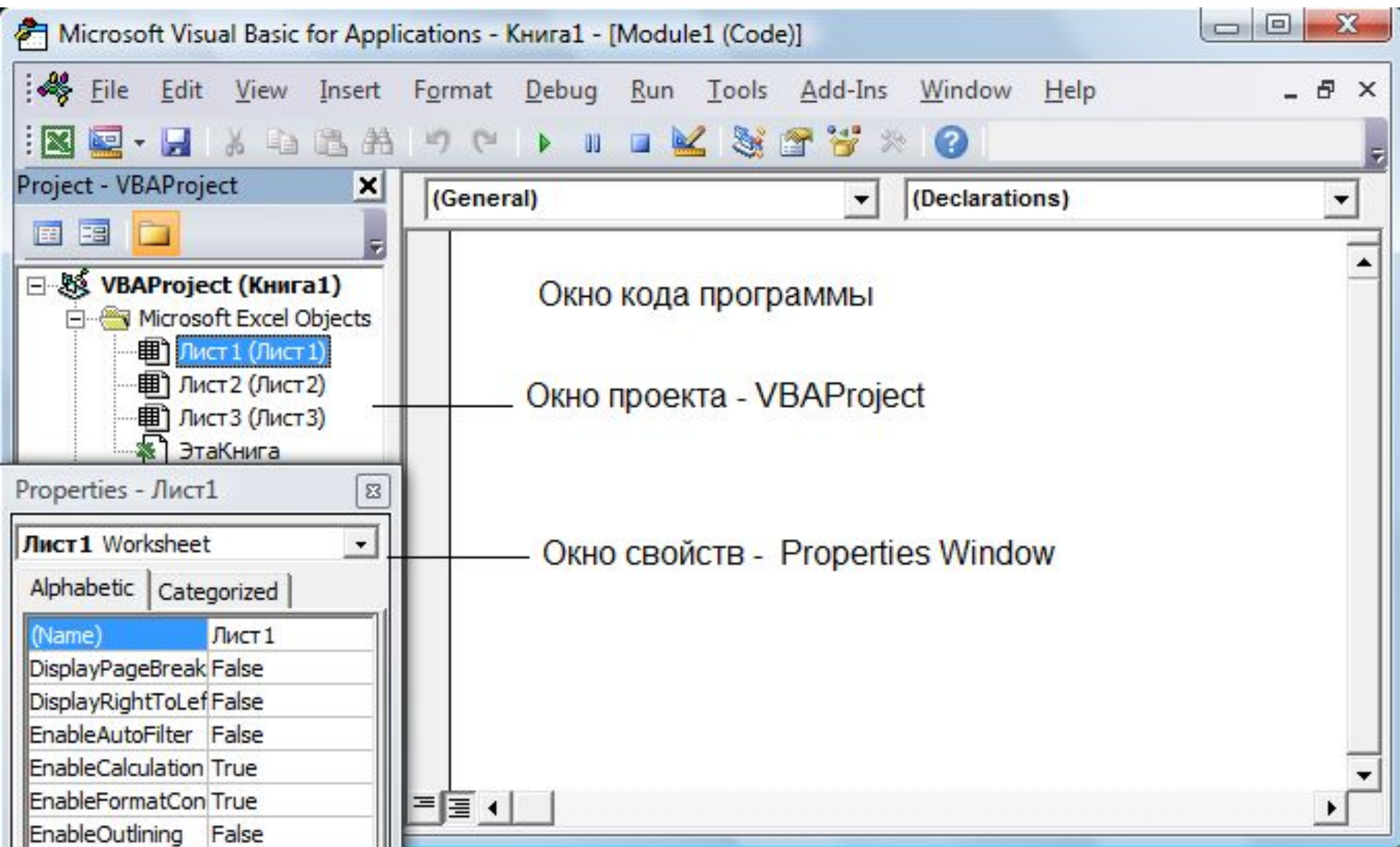


**Вкладка Разработчик**



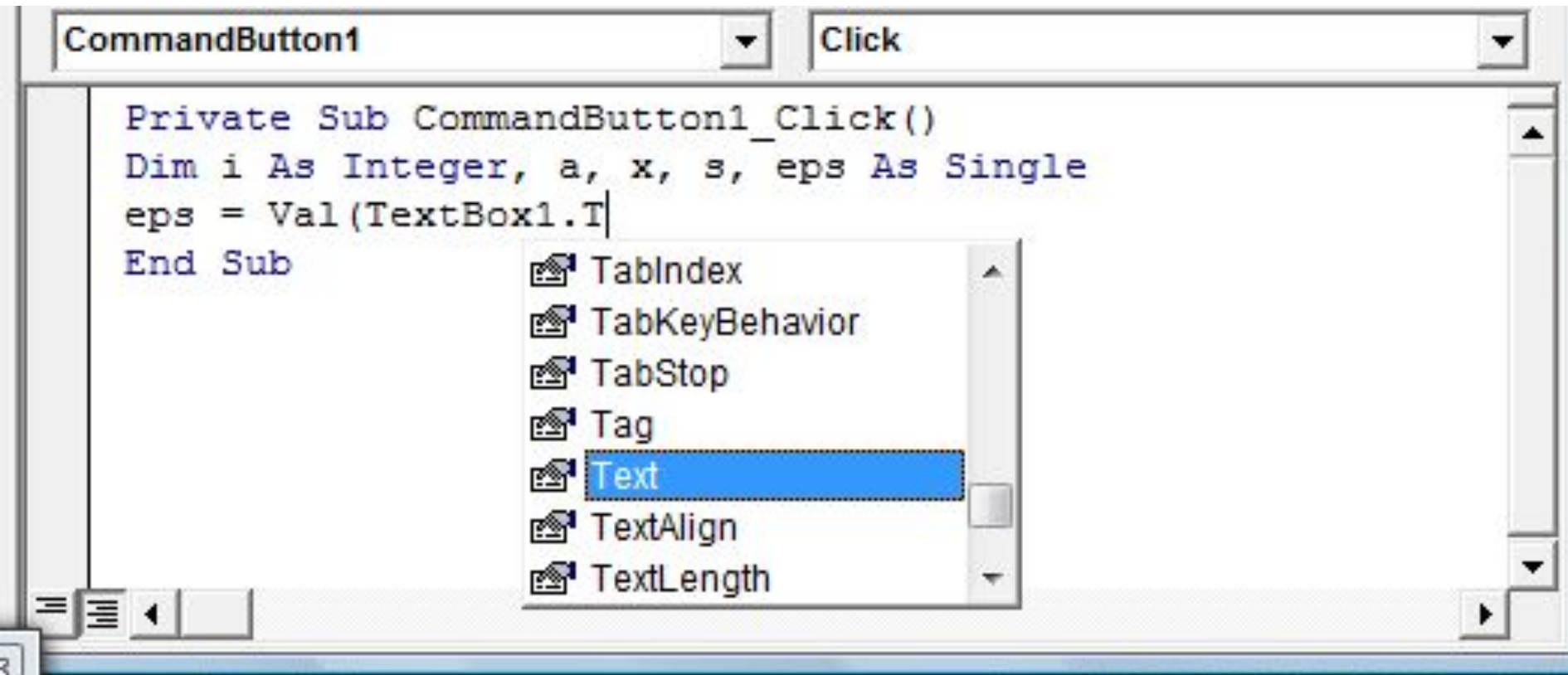
Окно Параметры EXCEL

# Редактор VBA



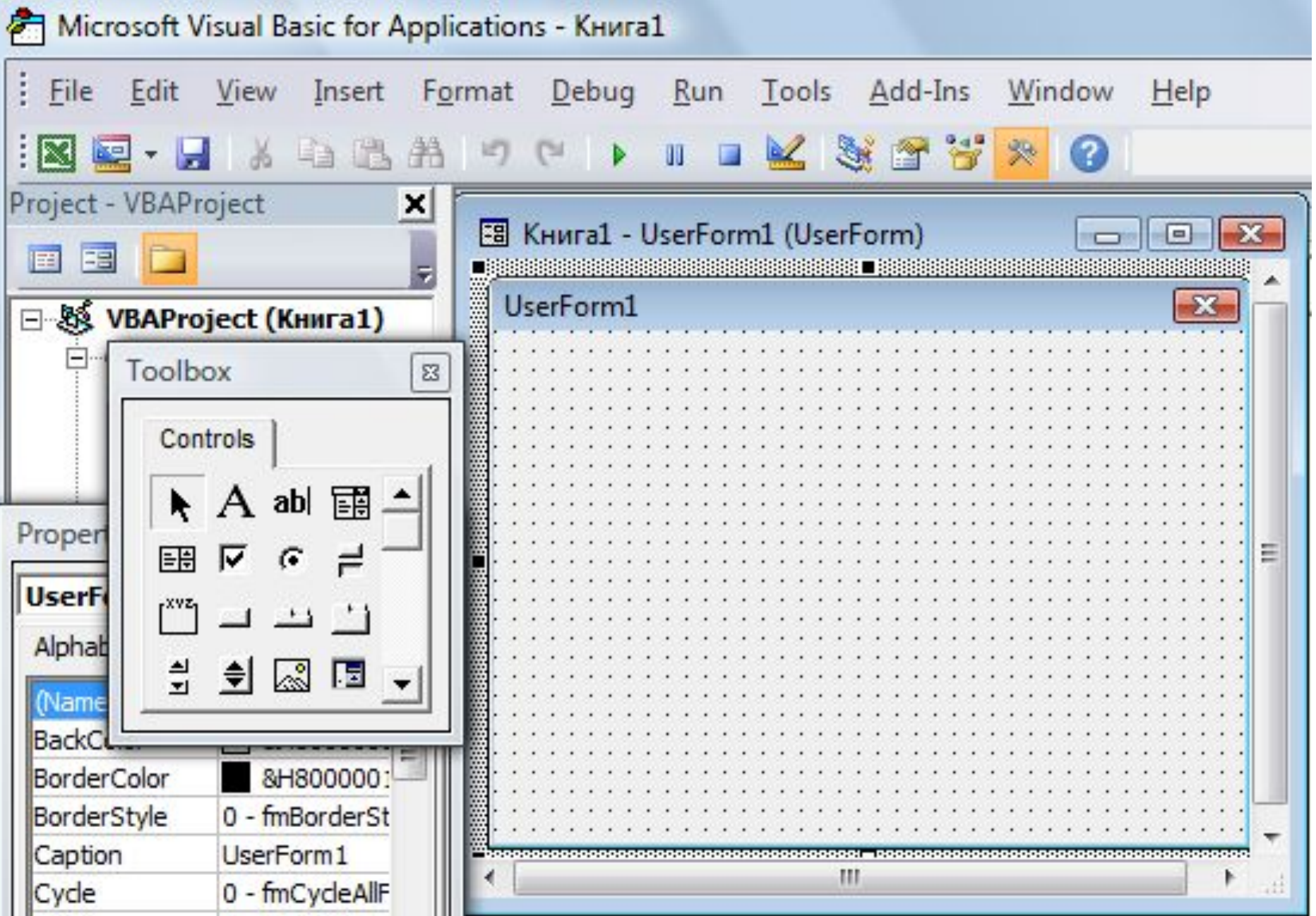
Окно редактора VBA





Автоматизированный ввод текста  
программы

# ФОРМЫ



Типы данных	байт	Диапазон значений
<b>Byte</b> (байт)	1	От 1 до 255
<b>Boolean</b> (логический)	2	True /False
<b>Integer</b> (целое число)	2	От -32 768 до 32 767
<b>Long</b> (длинное целое число)	4	От -2 147 483 648 до 2 147 483 647
<b>Single</b> (число с плавающей запятой обычной точности)	4	От -3,402823E38 до -1,401298E-45 для отрицательных значений; от 1,401298E-45 до 3,402823E38 для положительных значений
<b>Double</b> (число с плавающей запятой двойной точности)	8	От -1,7976931346232E308 до 1,7976931346232E308
<b>Currency</b> (денежный)	8	От -992 337 203 685 477,5808 до 992 337 203 685 477,5807
<b>String</b> (строка переменной длины) <b>String * длина</b>	длина строк и +10	От 0 до приблизительно 2 миллиардов От 0 до 65535 Строки фиксированной длины
<b>Object</b> (объект)	4	Любой указатель объекта
<b>Variant</b> по умолчанию		Может использоваться для хранения различных типов данных
Date дата	8	дробное число - целая часть количество дней с 31.12.1899, дробная часть - часы, минуты, секунды

Существует функция *VarType(Имя)*, которая возвращает число, соответствующее типу переменной.

0	1	2	3	4	5	6	7	8	9	11	17
пуст 0	NULL	Integer	Long	Single	Double	Currency	Date	String	Object	Boolean	Byte

```
Sub TestVar()
  Stemp = "Hello"
  MsgBox VarType(Stemp)
  Stemp = 4
  MsgBox VarType(Stemp)
End sub
```

*строки*  
инициализируются  
пустыми строками;  
числа - значением 0;  
переменные типа  
*Boolean* - False;  
даты - 30 декабря 1899

```
Dim a as Integer
a=InputBox("", "", "5")
‘ неявное преобразование типов a=5
```

```
Sub Proba1()
a=Inputbox("", "", "5") ‘variant в string
b=Inputbox("", "", "6") ‘variant в string
c=a+b
MsgBox c ‘ c=“56”
End sub
```

**Option Explicit** - требование явного  
объявления переменных  
(на уровне модуля)

## Описание переменных: [Public | Private | Dim | Static ] ИмяПеременной As Тип

1. Длина  $\leq 255$  символов.
2. Имя не может содержать точек, пробелов и следующих символов:  
%, !, &, #, @, \$.
3. Имя может содержать любую комбинацию букв, цифр, и символов, начинающуюся с буквы.
4. Имена должны быть уникальны в области, в которой они определены.
5. Не следует использовать имена, совпадающие с ключевыми словами VBA и именами встроенных процедур и функций.

## Константы

[Public | Private] Const ИмяКонстанты [As тип] = Выражение

## Область действия переменных

Переменные, описанные с помощью ключевого слова **Dim** на уровне модуля, доступны для всех процедур в данном модуле. Переменные, описанные на уровне процедуры, доступны только в данной процедуре. [**Public** | **Private**] используется на уровне модуля для описания констант (доступных всем модулям | только внутри модуля) .

# ОПЕРАЦИИ

## Математические операции:

+ - \* / \ Mod ^

## Операции отношения :

< <= > >= <> =

obj1 is obj2      str1 Like str2

## Логические операции:

And Or Xor Eqv Imp Not

## Математические функции:

Abs(x) Sqr(x) Tan(x) Atn(x) Cos(x)

Sin(x) Exp(x) Log(x) Sgn(x) Rnd(x)

Fix(x) Int(x)

## Функции преобразования

### ТИПОВ:

Str(n) Val(s) Asc(s) Chr(n)

CBool(a) CByte(a) CCur(a) CDate(a)

CDbl(a) CDec(a) CInt(a) CLng(a)

CSng(a) CVar(a) CStr(a)

**Оператор присваивания:** Переменная = выражение

## Перенос строки:

x = x + 1

или

x = x + 1 : y = x + 2

y = x + 2

## Комментарий:

x = x + 1 ' *Комментарий*

Rem *Комментарий*

**Окно сообщения** ( используется для вывода, диалога)

**MsgBox (Сообщение, Атрибуты, Заголовок, справка)**

**P=MsgBox(.....)**

**Сообщение** – выводимый текст.

**Атрибуты** = Параметр1 + Параметр2

Пара метр1	Набор кнопок
0	"ОК"
1	"ОК" и "Отмена"
2	"Прервать", "Повтор", "Пропустить"
3	"Да", "Нет", "Отмена"
4	"Да", "Нет"
5	"Повтор", "Отмена"

**Заголовок** — строка в заголовке окна сообщений.

**Справка** — имя файла справки

## Встроенные диалоговые окна

Пара метр2	Пиктограмма
16	
32	
48	
64	

**InputBox**(Message, Title , Default, X, Y, Info)

**Message** – Сообщение-подсказка в диалоговом окне

**Title**–надпись в строке заголовка окна

**Default**- значение по умолчанию в поле ввода

**X, Y** – координаты левого верхнего угла

P=InputBox(.....)

КНОПКИ	КОД	значения
vbOKOnly	0	OK
vbOKCancel	1	OK, Cancel
vbAbortRetryIgnore	2	Stop, Retry, Cancel
vbYesNoCancel	3	Yes, No, Cancel
vbYesNo	4	Yes, No
vbCritical	16	
vbQuestion	32	
vbExclamation	48	
vbInformation	64	

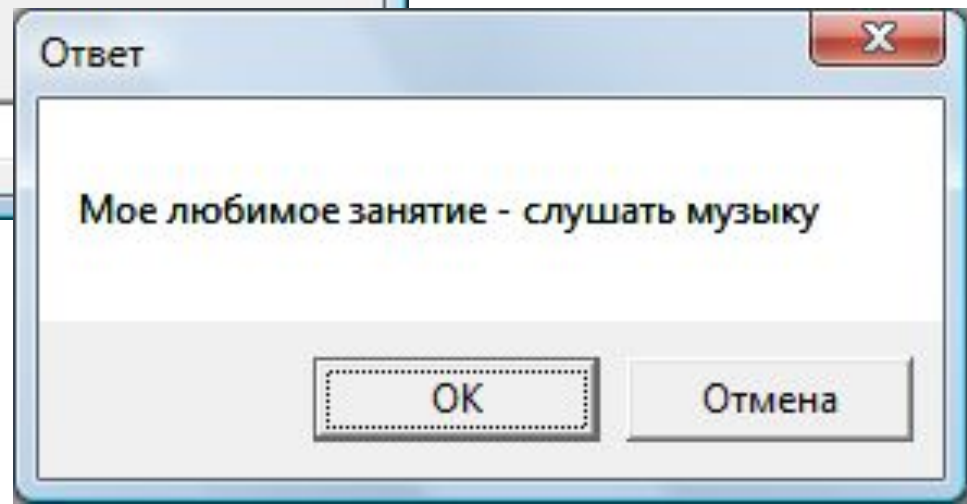
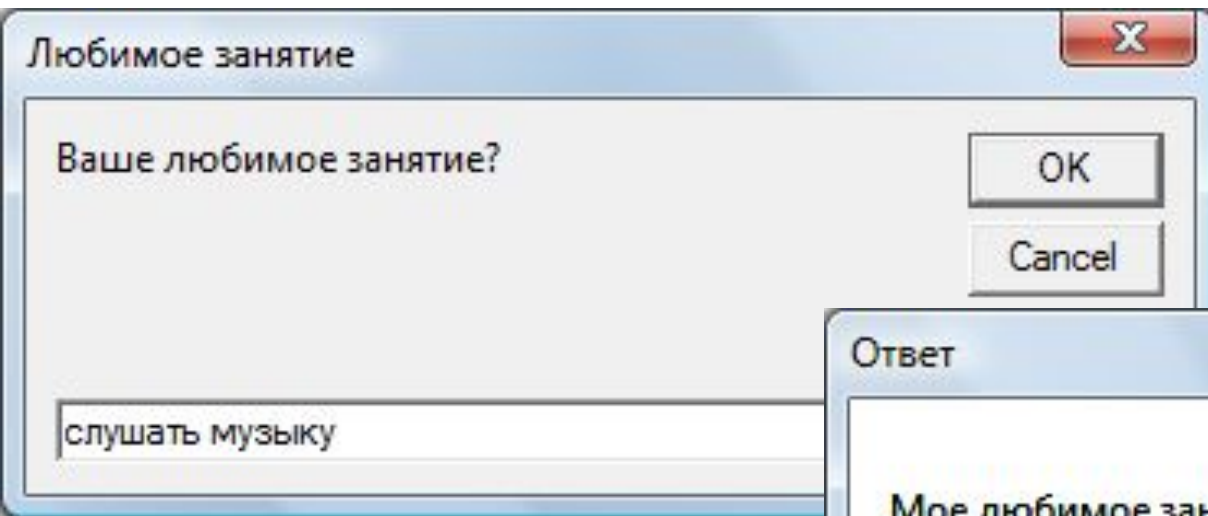
значения
vbOK
vbCancel
vbAbort
vbRetry
vbIgnore
vbYes
vbNo



```
MsgBox "Привет!", vbYesNoCancel + vbInformation , "Мое  
первое окно"
```

```
MsgBox "Первая программа", , "Окно сообщения"
```

```
Private Sub Example()  
Dim s As String  
s = InputBox("Ваше любимое занятие? ", "Любимое  
занятие")  
MsgBox "Мое любимое занятие - " & s, 1, "Ответ"  
End Sub
```



## Пример: Сумма двух чисел

```
Sub Summa()  
Dim A As Integer  
Dim B As Integer  
Dim C As Integer  
    A = Val(InputBox("A="))  
    B = Val(InputBox("B="))  
    C = A + B  
    MsgBox C  
End Sub
```

# **Управляющие алгоритмические конструкции**

**Конструкции ветвления (If и Select)**

**Циклические конструкции**

Однотрочная форма:

If U Then S1 Else S2

Блочная форма :

If U Then

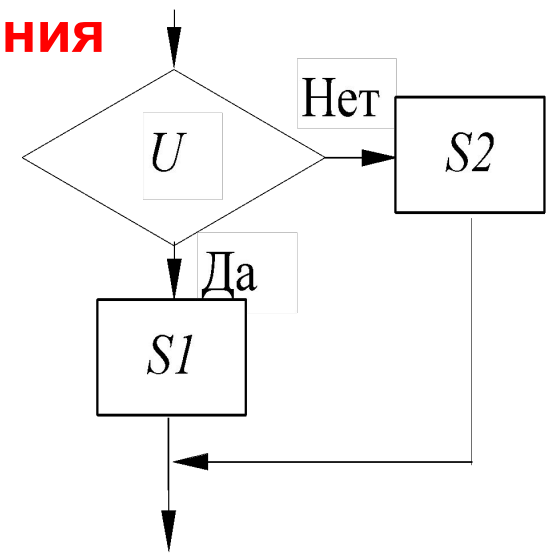
S1>

Else

S2>

End if

## Оператор ветвления



If A > 10 Then A = A + 1: B = B + A: C = C + B

If Alpha <=90 Then MsgBox "Острый угол" Else MsgBox "Тупой угол"

If x >= -5 And x < 5 Then K = K + 1: S = S + x

Sub Simple () ' определение четности числа

Dim X as integer

X=InputBox ("X=")

If X Mod 2 =0 then MsgBox "четное" else MsgBox "нечетное"

End sub

## ‘ Вложенные условные операторы

```
Sub School()  
dim Age as integer  
Age=InputBox(“Укажите возраст”)  
If Age >7 Then  
    If Age <=17 Then  
        MsgBox “Школьник”  
    Else  
        MsgBox “Взрослый”  
    End If  
Else  
    MsgBox “Дошкольник”  
End If  
End Sub
```

### Эквивалент вложенных операторов

```
If U1 Then  
S1  
Elseif U2 Then  
S2  
Elseif U3 Then S3  
Else  
S4  
End If
```

```
Sub School_1()  
Dim Age as integer  
Age=InputBox(“Укажите возраст”)  
If Age <7 Then  
    MsgBox “Дошкольник”  
    ElseIf Age <=7 Then  
        MsgBox “Школьник”  
    ElseIf Age <23 Then  
        MsgBox “Студент”  
    ElseIf Age <55 Then  
        MsgBox “Специалист”  
    Else  
        MsgBox “Пенсионер”  
    End If  
End Sub
```

# SELECT CASE

```
Select Case(Grade)
Case 1
....
Case 2,3
....
Case 4 to 6
....
Case Is > 8
....
Case Else
...
End Select
```

```
Sub Scase()
Dim a as Integer
A=InputBox("Ваш рост в см")
Select Case A
Case Is <160
MsgBox "У вас маленький рост"
Case 161 to 180
MsgBox "Вы человек среднего роста"
Case Is >180
MsgBox "Вы высокий человек"
End Select
End Sub
```

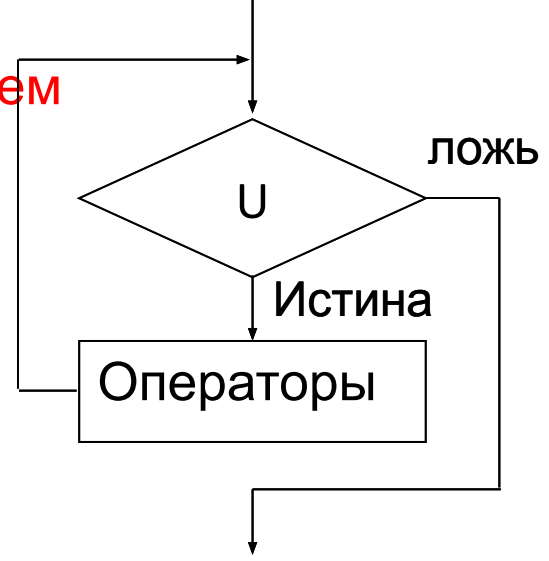
# Циклы

- С предусловием;
- С постусловием

# ЦИКЛЫ

## ЦИКЛ с предусловием

**While U**  
**Операторы**  
**Wend**

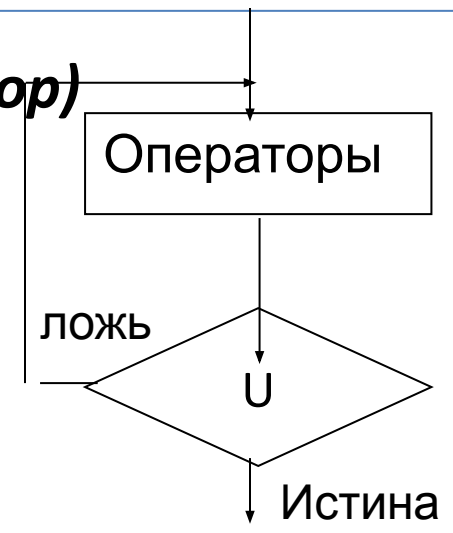


```
Sub WH_1()
Dim N as integer, k as integer
K=0
While N>0
K=K+N Mod 10
N=N \ 10
Wend
MsgBox "Сумма цифр =" & k
End Sub
```

## Операторы цикла с условиями (Do ... Loop)

**С предусловием**  
**Do While | Until U**  
**Операторы**  
**[Exit Do]**  
**Loop**

**С постусловием**  
**Do**  
**Операторы**  
**[Exit Do]**  
**Loop While | Until U**



<pre>k = 0 Do While N &gt; 0   k = k + 1   N = N \ 10 Loop</pre>	<pre>k = 0 Do Until N = 0   k = k + 1   N = N \ 10 Loop</pre>	<pre>k = 0 Do   k = k + 1   N = N \ 10 Loop While N &gt; 0</pre>	<pre>k = 0 Do   k = k + 1   N = N \ 10 Loop Until N = 0</pre>
--	---	--	---



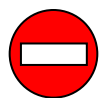
# Вычисление НОД

---

**НОД** = наибольший общий делитель двух натуральных чисел – это наибольшее число, на которое оба исходных числа делятся без остатка.

Перебор:

```
k = a \ или k = b;  
while ( a Mod k <> 0 Or b Mod k <> 0 )  
    k =k-1  
Wend  
MsgBox "НОД=" & k
```



много операций для больших чисел

# Алгоритм Евклида

---

$$\begin{aligned}\text{НОД}(a, b) &= \text{НОД}(a-b, b) \\ &= \text{НОД}(a, b-a)\end{aligned}$$

Заменяем большее из двух чисел **разностью** большего и меньшего до тех пор, пока они не станут равны. Это и есть НОД.



Евклид  
(365-300 до. н. э.)

**Пример:**

$$\begin{aligned}\text{НОД}(14, 21) &= \text{НОД}(14, 21-14) = \text{НОД}(14, 7) \\ &= \text{НОД}(7, 7) = 7\end{aligned}$$

 много шагов при большой разнице чисел:

$$\text{НОД}(1998, 2) = \text{НОД}(1996, 2) = \dots = 2$$

# Модифицированный алгоритм Евклида

---

$$\begin{aligned}\text{НОД}(a, b) &= \text{НОД}(a \bmod b, b) \\ &= \text{НОД}(a, b \bmod a)\end{aligned}$$

Заменяем большее из двух чисел **остатком от деления** большего на меньшее до тех пор, пока меньшее не станет равно нулю. Тогда большее — это НОД.

**Пример:**

$$\text{НОД}(14, 21) = \text{НОД}(14, 7) = \text{НОД}(0, 7) = 7$$

**Еще один вариант:**

$$\text{НОД}(2 \cdot a, 2 \cdot b) = 2 \cdot \text{НОД}(a, b)$$

$$\text{НОД}(2 \cdot a, b) = \text{НОД}(a, b) \quad // \quad \text{при нечетном } b$$

# Реализация алгоритма Евклида

---

```
Sub NOD1 ()
Dim a as integer
Dim b as integer
while ( a <> b )
    if ( a > b ) Then
        a = a - b
    else
        b = b - a
    End if
wend
MsgBox "Nod=" & a
End Sub
```

```
Sub NOD2 ()
Dim a as integer
Dim b as integer
while ( a*b <> 0 )
    if ( a > b ) Then
        a = a Mod b
    else
        b = b Mod a
    MsgBox "Nod=" & (a + b)
End Sub
```



Почему MsgBox (a+b) ?

# ЧИСЛОВЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ $x_1, x_2, \dots, x_k, \dots$

**Последовательность** — это набор элементов, расположенных в определенном порядке.

Все элементы последовательности имеют номера, обычно начиная с 1. Для того, чтобы задать последовательность, используют два следующих способа.

1. **Рекуррентную формулу**, которая позволяет вычислить элемент с номером  $n$ , зная один или несколько предыдущих. Например

последовательность

1, 3, 5, 7, ... можно задать рекуррентной формулой

2. **Формулу для  $n$ -ого члена последовательности**. Для той же последовательности легко получить

. Правая часть этой формулы зависит только от номера элемента  $n$ .

Для хорошо известной арифметической прогрессии формула  $n$ -ого члена имеет вид

**Для геометрической прогрессии, соответственно,**

## Обработка

## последовательностей

 $x_1, x_2, \dots, x_k, \dots$ 

При работе с последовательностями используют конечное количество



ячеек, не зависящее от количества элементов в последовательности.

**Найти  
максимальный  
элемент  
последовательности  
ненулевых  
целых чисел.  
0 - признак  
конца  
последовательности.**

```
Sub Max_SQ()  
Dim x as integer, max as integer  
x=InputBox("x=") : max=x  
While x<>0  
If max >x then max=x  
x=InputBox("x=")  
Wend  
If max=0 Then  
MsgBox "пусто"  
Else  
MsgBox "max=" & max  
End if  
End Sub
```

# Рекуррентные последовательности

В 1202 г. итальянский математик Леонардо Пизанский, известный под именем Фибоначчи, предложил такую задачу:

**Задача Фибоначчи.** Пара кроликов каждый месяц дает приплод – самца и самку, которые через 2 месяца снова дают такой же приплод. Сколько пар кроликов будет через год, если сейчас мы имеем 1 пару молодых кроликов?

Количество кроликов меняется с каждым месяцем в соответствии с последовательностью **1, 1, 2, 3, 5, 8, 13, 21, 34, ...**, которую называют **последовательностью Фибоначчи**. Она задается **не общей формулой n-ого члена, а рекуррентной формулой, в которой n-ый член выражается через предыдущие**. При этом надо определить начальные элементы:

$$f_1 = 1, f_2 = 1,$$

$$f_n = f_{n-1} + f_{n-2}, \quad n > 2$$

```
n=InputBox("n=")
f1=1: f2=1: k=2
While k<n
f1=f1+f2 : f2=f1-f2:
k=k+1
Wend
MsgBox f1
End Sub
```

*Получить все числа Фибоначчи, не превышающие данного N.*

```
Sub Fibb()  
Dim N as integer  
Dim f1 as integer, f2 as integer  
f1=1 : f2=1  
While f1 <= N  
    MsgBox f1  
    f1=f1 +f2  
    f2=f1-f2  
Wend  
End Sub
```

**Самостоятельно:**

1. Найти наименьший элемент Фибоначчи, больший данного натурального N.
2. Найти сумму четных элементов Фибоначчи, не превышающих данного N.



# Найти сумму элементов последовательности целых чисел, номера которых являются элементами Фибоначчи.

**0 – признак конца последовательности.**

```
Sub Sum_Fib()  
Dim x as integer, S as integer  
Dim n as integer  
Dim f1 as integer, f2 as integer  
x=InputBox("x="): S=0: f2=1 : f1=1: n=1  
While x<>0  
If n=f1 then S=S+x: f1=f1+f2: f2=f1-f2  
x=InputBox("x="): n=n+1  
Wend  
MsgBox "S=" & S  
End Sub
```

X:	2	5	-6	8	45	-89	7	98	90	-4	9	56	5	...
n:	1	2	3	4	5	6	7	8	9	10	11	12	13....	
f:	1	2	3	-	5	-	-	8	-	-	-	-	13	....
f2	f1													
	f2	f1												
		f2	f1											
			f2	-	f1									
				f2	-	-	f1							