

Учебный курс  
**«ИНФОРМАТИКА»**

---

Преподаватель:  
ст. преп. Зуева Екатерина Александровна

# **Visual Basic for Applications (VBA)**

---

Лекция 10

# Visual Basic for Applications (VBA)

---

1. Конструкции языка.
2. Основные понятия.
3. Интерфейс.
4. Объекты.
5. Работа с объектами, события, методы.

# Visual Basic for Applications (VBA)

---

Visual Basic for Applications (VBA, Visual Basic для приложений) - немного упрощённая реализация языка программирования Visual Basic, встроенная в линейку продуктов Microsoft Office (включая версии для Mac OS), а также во многие другие программные пакеты, такие как AutoCAD, SolidWorks, CorelDRAW, WordPerfect и ESRI ArcGIS.

VBA является интерпретируемым языком. Как и следует из его названия, VBA близок к Visual Basic. VBA, будучи языком, построенным на COM, позволяет использовать все доступные в операционной системе COM объекты и компоненты ActiveX.

# Visual Basic for Applications (VBA)

К достоинствам языка можно отнести:

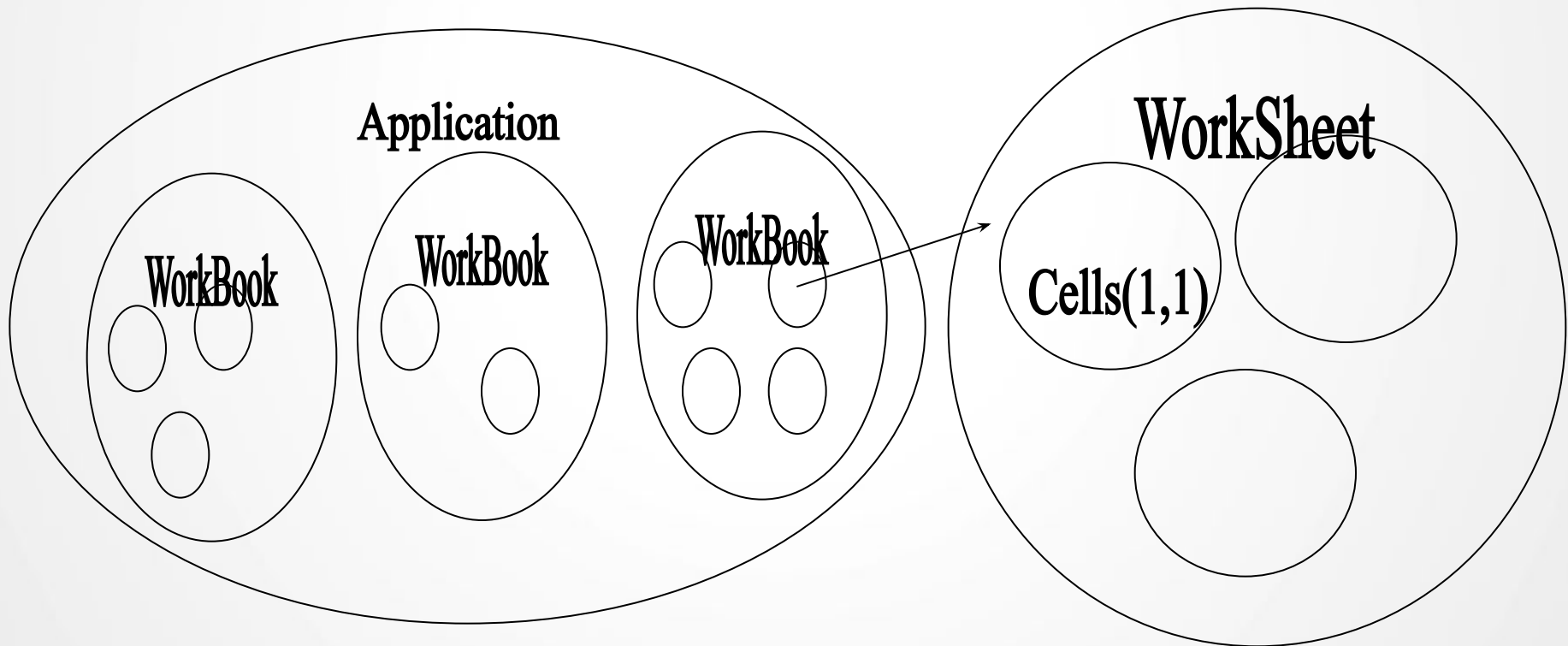
- сравнительную лёгкость освоения (приложения могут создавать даже новички);
- выполнение скрипта именно в среде офисных приложений;
- создавать автоматизированные информационные системы;
- легко разрабатывать пользовательский интерфейс.

Недостатком являются проблемы с обратной совместимостью разных версий.

Программирование — создание программ. Офисное программирование — создание документов и программа является лишь его частью. Все создаваемые программные компоненты документа объединены в одно целое, называемое **проектом** — совокупность программных модулей и форм.

# Visual Basic for Applications (VBA)

Приложения из состава Office - это совокупность вложенных объектов. Объектами являются сами приложения и все их компоненты. Примерная структура Excel (более 100 объектов):



# Структура объектов

Свойства – описывают состояние объекта;

Методы – определяют поведение объекта;

События – определяют то, что может происходить с объектом.

Например: кнопка



Свойство - Цвет кнопки;

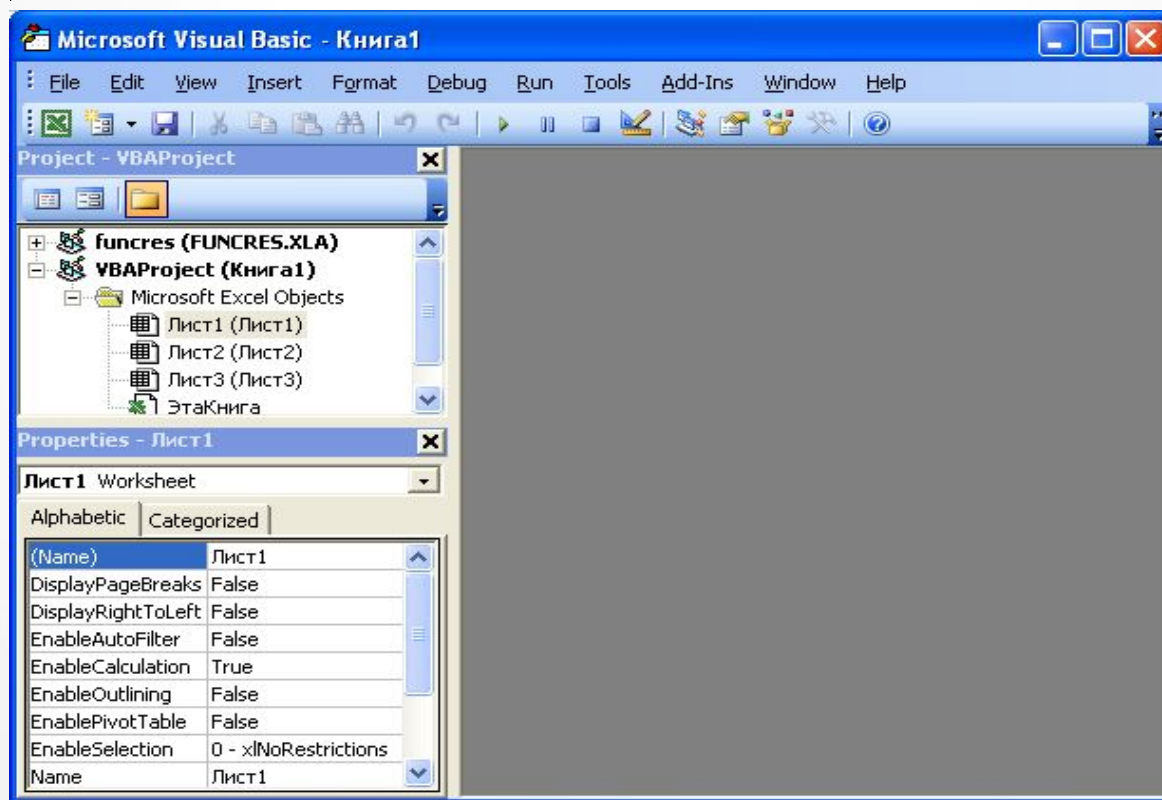
Метод - Пользователь может нажимать на кнопку мышью;

Событие - Надпись на кнопке изменяется при нажатии на нее.

# Интегрированная среда разр-ки

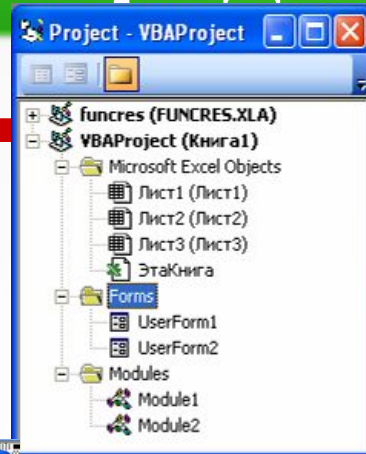
Для перехода в редактор:

1. команда Сервис–Макрос–Редактор Visual Basic;
2. комбинация клавиш “Alt + F11”.

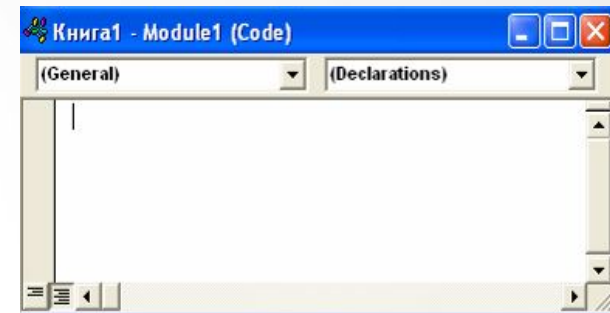




# Структура редактора VBA

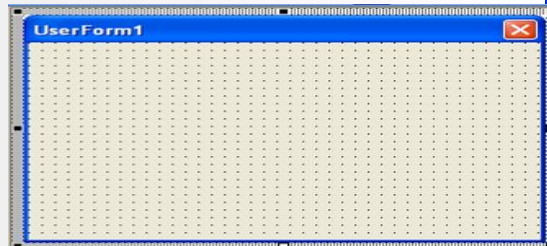


Окно проекта (Project);



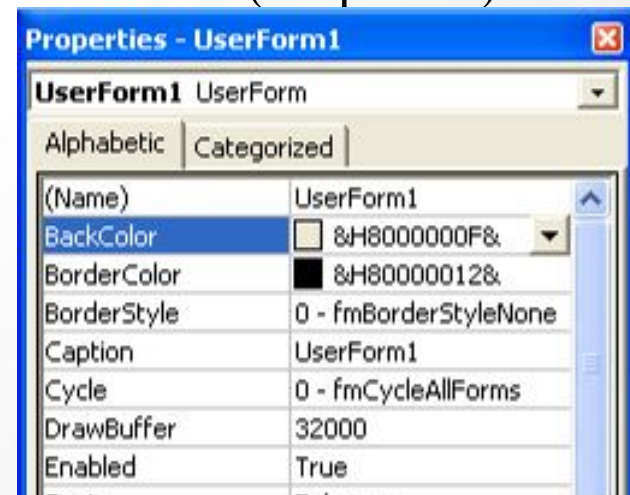
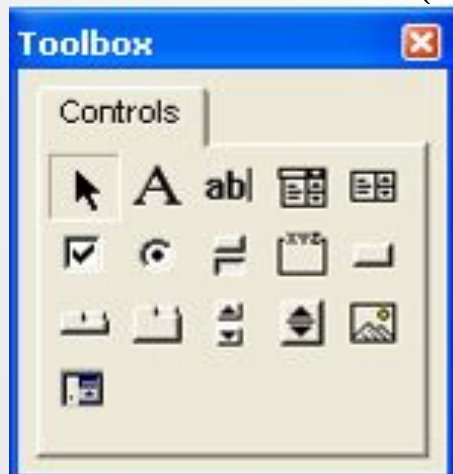
Окно редактирования кода (Code);

Окно редактирования форм (UserForm);



Панель элементов (ToolBox);

Окно свойств (Properties).



# Возможности редактора

После нажатия клавиши “Enter” красным цветом выделяется та строка в программе, в которой допущена синтаксическая ошибка. Также на экране отображается диалоговое окно, поясняющее, какая ошибка могла произойти.

```
Private Sub UserForm_Click()
```

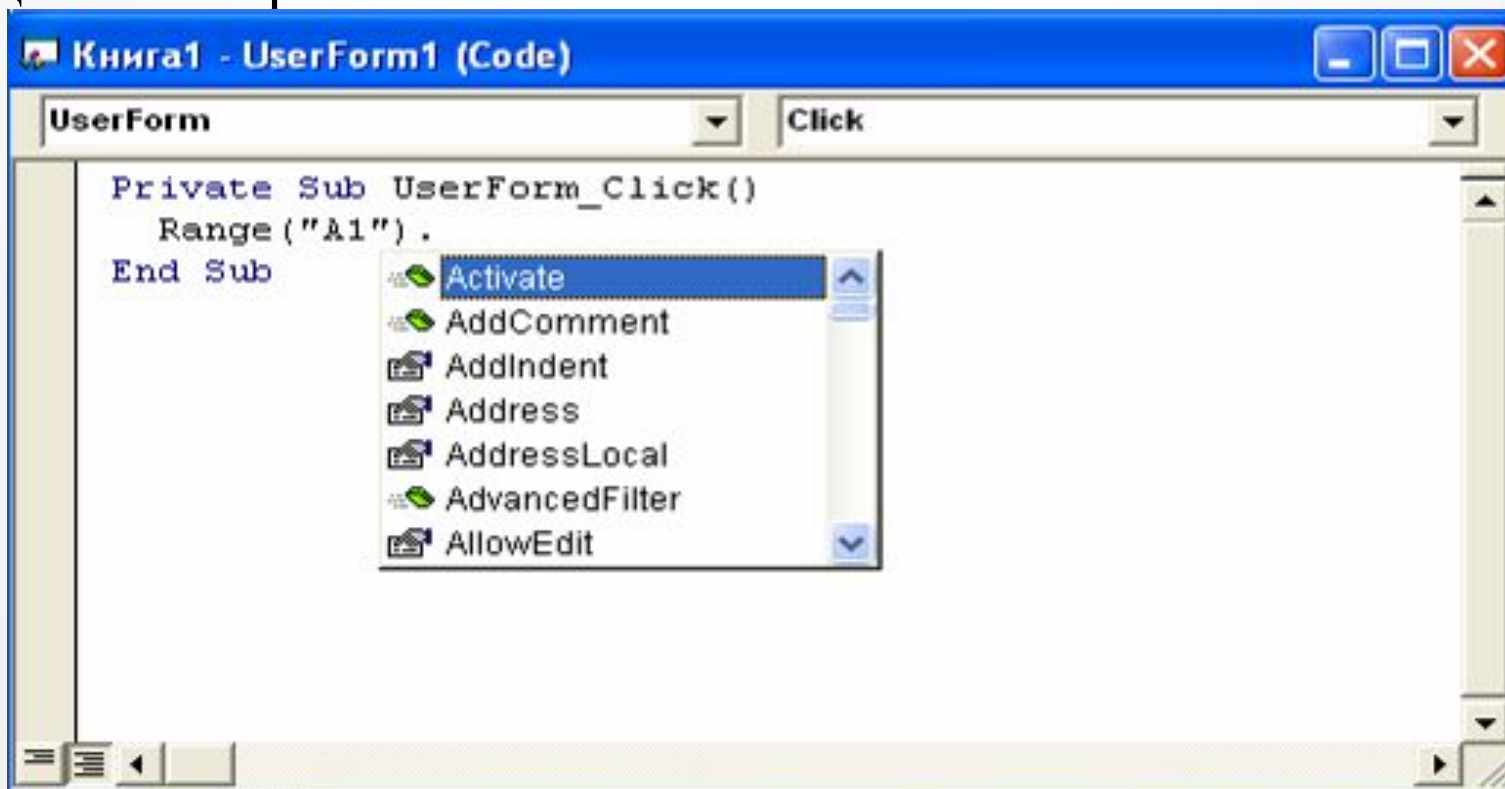
```
    Caption = "Проек"
```

```
    Caption := "Проек"
```

```
End Sub
```

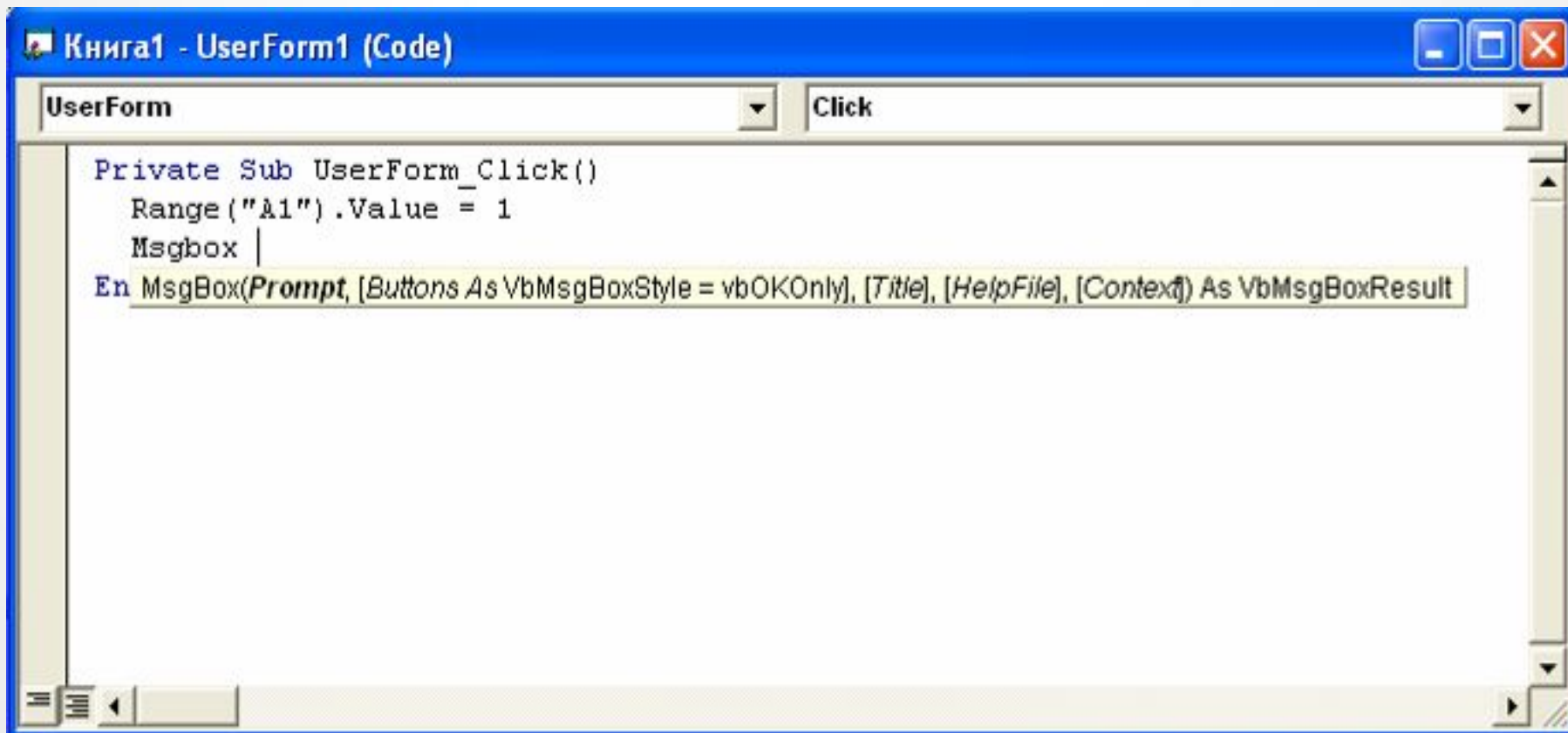
# Возможности редактора

Выводится список компонентов, логически завершающих вводимую инструкцию. Нажатие клавиши “Tab” вставляет выбранное имя в код программы. Список компонентов можно выводить на экран нажатием “Ctrl+J”.



# Возможности редактора

Отображаются сведения о процедурах, функциях, свойствах и методах после набора их имени. Всплывающую подсказку можно также выводить на экран нажатием “Ctrl+I”.



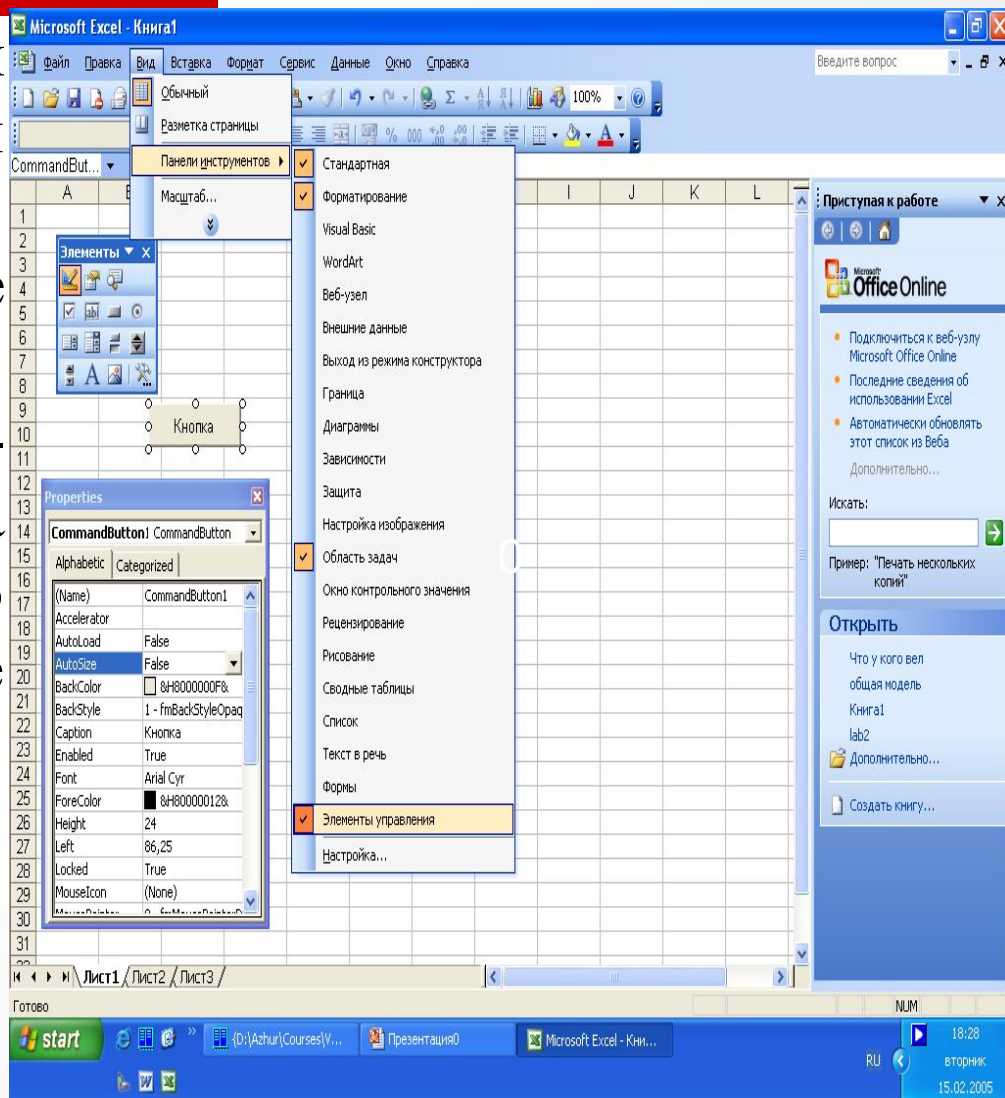
The screenshot shows a code editor window titled "Книга1 - UserForm1 (Code)". The editor displays a VBA subroutine named "UserForm\_Click()". The code is as follows:

```
Private Sub UserForm_Click()  
    Range("A1").Value = 1  
    MsgBox |  
End Sub
```

A tooltip is visible over the "MsgBox" line, displaying the function signature: `MsgBox(Prompt, [Buttons As VbMsgBoxStyle = vbOKOnly], [Title], [HelpFile], [Context]) As VbMsgBoxResult`. The tooltip text is partially obscured by a vertical bar.

# Возможности редактора

При помощи панели инструментов Элементы управления, которая отображается на экране выбором команды Вид-Панели инструментов-Элементы управления, на рабочем листе можно размещать различные элементы.



# Возможности редактора

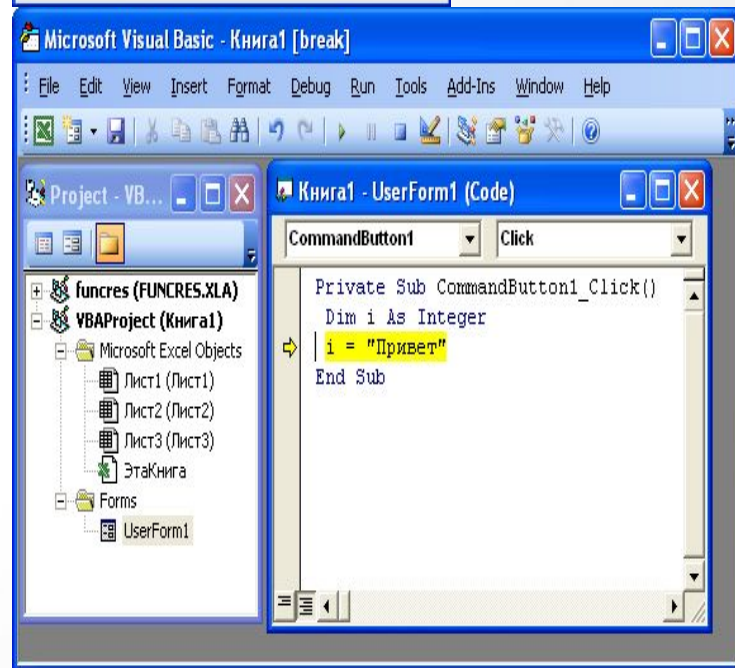
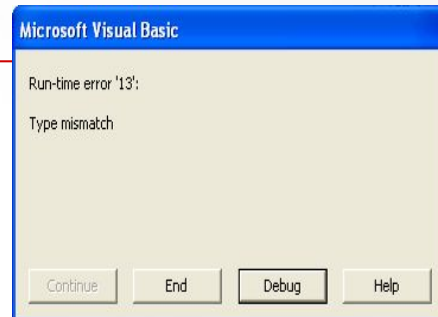
При возникновении ошибки, во время работы программы, выводится диалоговое окно с пояснением.

## Действия:

Нажать на кнопку “Debug” для исправления ошибки или “End” для остановки выполнения программы;

Желтым цветом выделяется та строка, в которой допущена ошибка;

Исправить ошибку и выполнить команду Run-Continue или нажать на кнопку для продолжения выполнения программы.



# Основные элементы управления

## 1 НАДПИСЬ (Label)

- **Name** - имя элемента управления
- **Caption** (заголовок) - задает заголовок элемента на форме
- **Font** (шрифт) задает шрифт, которым должен быть написан заголовок.
- **FillColor** – задает цвет заполнения области.
- **ForeColor** - используется для установки цвета шрифта.
- **BackColor** - устанавливает цвет фона.
- **Alignment** (выравнивание) - позволяет установить тип выравнивания для записи.
- **AutoSize** – позволяет автоматически устанавливать ширину поля в зависимости от числа находящихся в нем символов. Если присвоить этому полю значение True, то ширина поля будет зависеть от количества введенных символов.
- **BorderStyle** – используется для задания стиля рамки
- **WordWrap** – значение True определяет перенос текста внутри поля по словам.

# Текстовое поле (TextBox)

- **Name** - задает имя текстового поля (по умолчанию Text1)
- **Text** - определяет содержимое текстового поля.
- **MaxLength** - ограничивает длину вводимого текста заданным количеством символов.
- **MultiLine** (True) - позволяет создать многострочное текстовое поле и новый абзац при нажатии клавиши <Enter>.
- **ScrollBars** – устанавливает полосы прокрутки для перемещения по тексту (1 - горизонтальная, 2 - вертикальная).
- **Index** – со значением 0 задает размножаемое текстовое поле.



# Командная кнопка (CommandButton)

Как правило, работа с кнопками заключается в установке их свойств и написании кода для события Click (Щелчок).

- **Name** – имя элемента управления
- **Caption** – текст, который появится на кнопке.

Если свойство **Style** (Стиль) имеет значение **Graphical** (Графический), то можно разместить рисунок на кнопке.

При этом в значении свойства **Picture** (Рисунок) требуется подобрать подходящее растровое изображение.

## Типы данных языка

- **Integer** (2 байта) – целое число из диапазона -32768 ...32767
- **Long** (4 байта) – целое число из диапазона -2147483648...2147483647
- **Single** (4 байта) – вещественное число из диапазона для: отрицательных –  $-3,4e38$  ...  $-1,4e-45$ , для положительных –  $1,4e-45$  до  $3,4e38$
- **Double** (8 байтов) – вещественное число с удвоенной длиной из диапазона -  $4,94e-324$  до  $1,797e308$
- **Currency** (8 байтов) – денежные величины

## Типы данных языка

- **String** (10 байтов + длина переменной) – текстовые или строковые значения. Строки переменной длины могут содержать до миллиарда символов, а строки фиксированной длины – до 65400 символов
- **Byte** (1 байт) – целое число от 0 до 255
- **Boolean** (2 байта) – логическое значение True (1) или False (0)
- **Date** (8 байтов) – даты с 1 января 100 года до 31 декабря 9999 года
- **Variant** – данные любого типа (назначена по умолчанию)

# Объявление данных

Имя переменной или константы начинается с буквы и регистр букв не имеет значения .

## 1.1 Объявление констант

:

**Const** *Имя константы* [**As тип**] = значение

**Const Pi As Double = 3.14159**

**Const Pi = 3.14159**

## 1.2 Объявление переменных

**Dim** *Имя переменной* **As тип**

**Dim d As Single, строка As String, str As String\*4**

# Знаки операций

Действие	Знак	VB
Возведение в степень	$\wedge$	$a \wedge b$
Умножение	$*$	$a * b$
Остаток от деления на целое	$/$	$a / b$
Остаток от деления на целое	Mod	$a \text{ Mod } b$
Сложение	$+$	$a + b$
Вычитание	$\square$	$a \square b$

# Арифметические функции

Результат	Обозначение	Функция
Абсолютное значение	$ x $	Abs(x)
Квадратный корень	$\sqrt{x}$	Sqr (x)
Натуральный логарифм	$\ln x$	Log (x)
Показательная функция	$e^x$	Exp (x)
Возведение в степень	$x^3$	$x^3$

# Тригонометрические функции

Результат	Обозначение	Функция
Синус угла	$\sin x$	$\sin(x)$
Косинус угла	$\cos x$	$\cos(x)$
Тангенс угла	$\operatorname{Tg} x$	$\tan(x)$
Арктангенс угла	$\operatorname{Arctg} x$	$\operatorname{Atn}(x)$

*радианная мера = градусная мера \*  $\pi / 180$*   
*градусная мера = радианная мера \*  $180 / \pi$*

Логарифм по заданному основанию

$$\text{Log}_5 X = \log(x) / \log(5)$$



# Организация ввода данных

## 1. С помощью текстового поля

Для организации ввода значения с помощью текстового поля необходимо выполнить следующие действия:

– на форме разместить текстовое поле для ввода значения;

– в программе записать оператор

*имя переменной = Val (Text1.Text)*

или

*имя переменной =Text1*

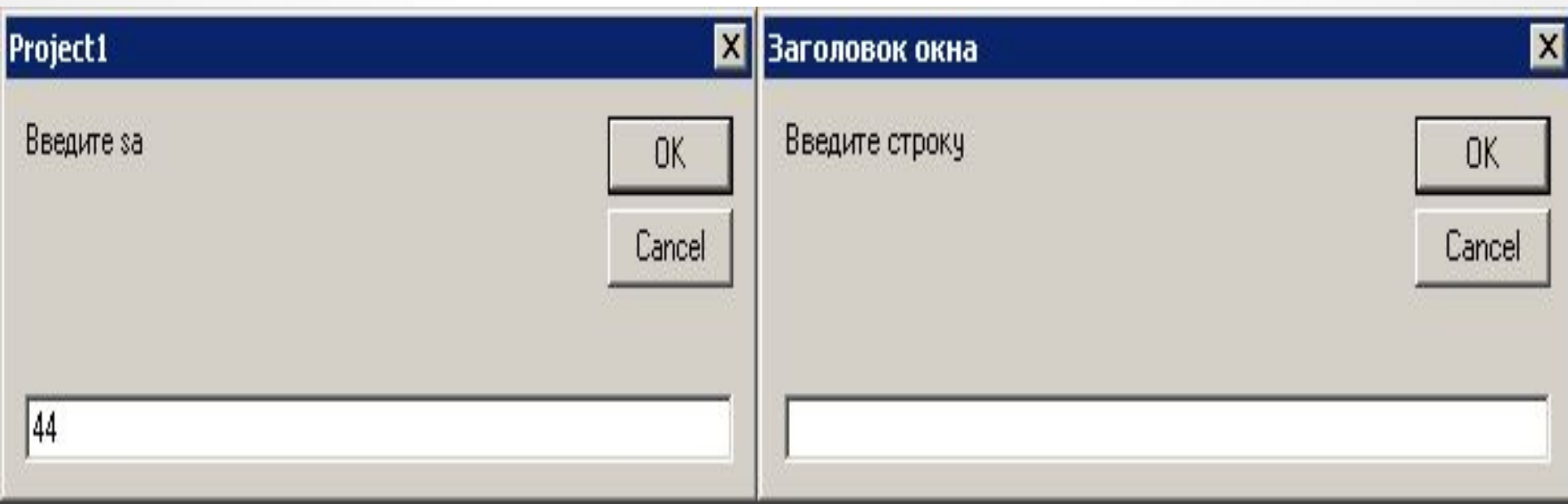
## 2. С помощью функции InputBox()

*Имя переменной = InputBox ("сообщение", "  
заголовок")*

*[,умолчание][,pos1, pos2])*

*sa = InputBox(" Введите sa",,44)*

*Строка= InputBox("Введите строку", "  
Заголовок окна")*



# Организация вывода данных

## 1. С помощью текстового поля

`Text2.Text=a` или `Text2=a`

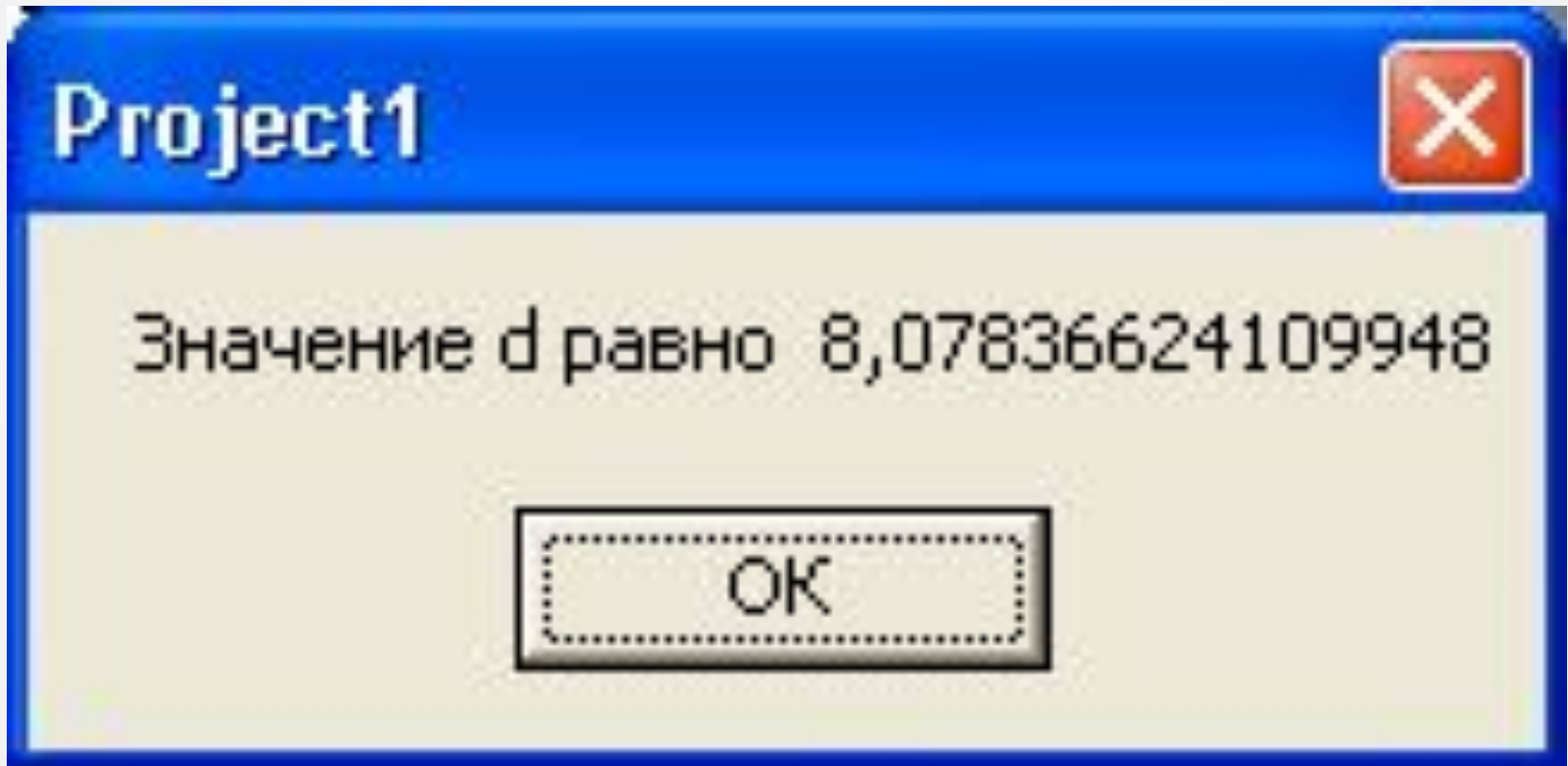
## 2. С помощью оператора MsgBox

`MsgBox "сообщение" [, атрибуты][, "заголовок"]`

- *сообщение* – текст, отображаемый в диалоговом окне (текст должен быть заключен в двойные кавычки);
- *заголовок* – текст, который размещается в заголовке окна;
- *атрибут* – значения этого параметра определяют, какие кнопки должны отобразиться в окне.

`MsgBox "Значение d =" & d`

# Окно вывода оператора MsgBox



# Значения параметра атрибут функции MsgBox()

Константа	Знач-е параметра	Отображаемые кнопки
VbOKOnly	0	
VbOKCancel	1	 
VbYesNoCancel	3	  
VbYesNo	4	 
VbQuestion	32	
VbDefaultButton1	0	По умолч-ю активна 1-ая кнопка  
VbDefaultButton2	256	По умолч-ю активна 2-ая кнопка

## Функция *MsgBox()*

Выводит данные в стандартное диалоговое окно, которое может иметь текст сообщения, заголовок и набор кнопок.

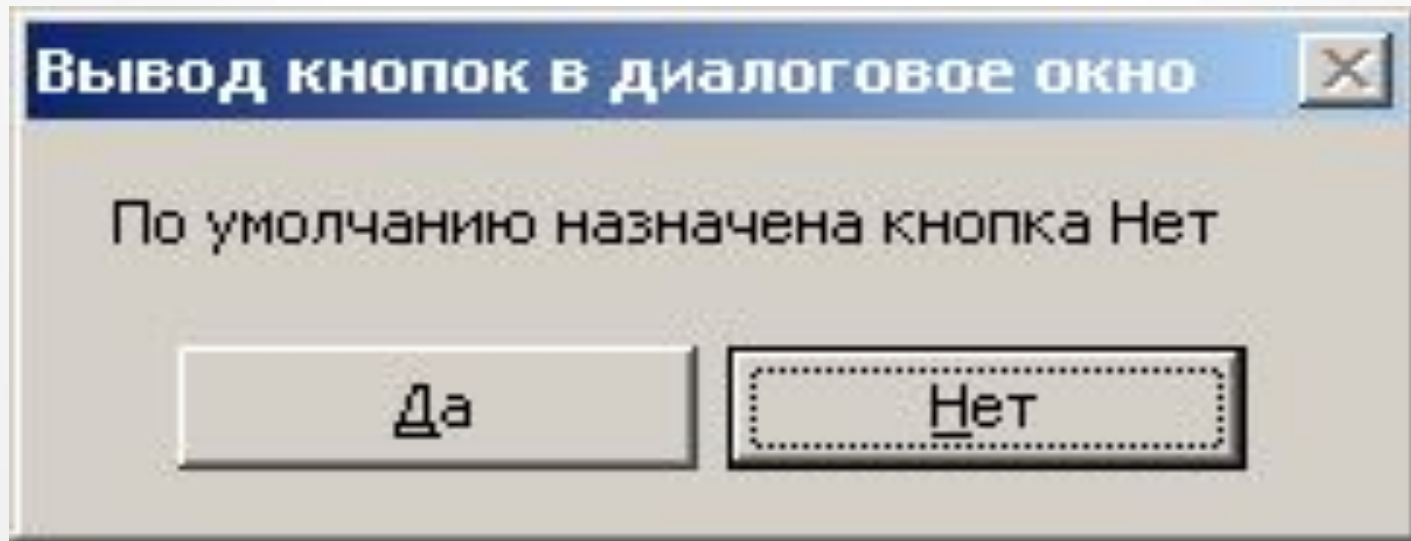
Синтаксис функции вывода сообщений в стандартное диалоговое окно:

**имя переменной = MsgBox(сообщение, атрибут, заголовок)**

В отличии от оператора *MsgBox* функция возвращает значение параметра *атрибут*, позволяющее определить, какую кнопку нажал пользователь в стандартном диалоговом окне.

MsgBox "По умолчанию назначена кнопка Нет", \_  
vbYesNo + vbDefaultButton2, "Вывод кнопок в \_  
диалоговое окно"

MsgBox "По ум-ю назначена кнопка Нет", 4 +256,  
—  
"Вывод кнопок в диалоговое окно"



### 3. С помощью функции `MsgBox()`

Выводит данные в стандартное диалоговое окно, которое может иметь текст сообщения, заголовок и набор кнопок.

Синтаксис функции вывода сообщений в стандартное диалоговое окно:

```
имя переменной = MsgBox(“сообщение”[,  
атрибут][,“заголовок”])
```

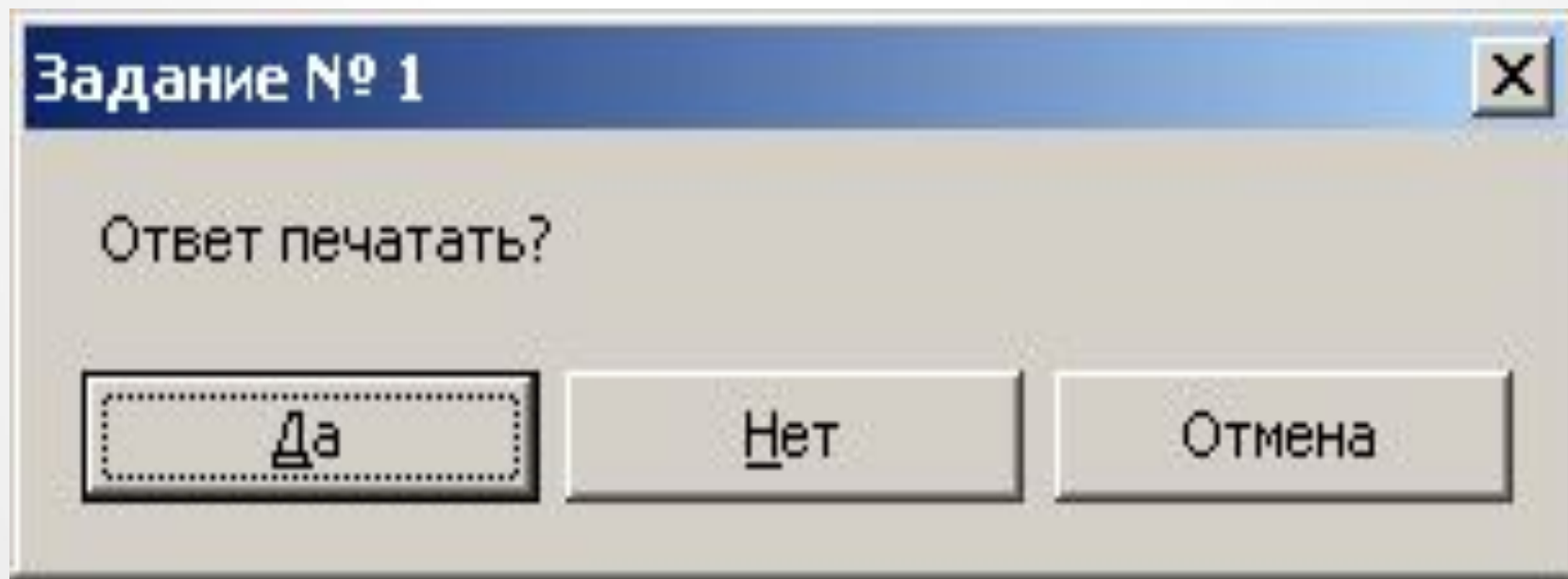
В отличие от оператора `MsgBox` функция возвращает значение, позволяющее определить, какую кнопку нажал пользователь в диалоговом окне.



# Значение параметра атрибут возвращаемое функцией MsgBox()

Константа	Значение параметр а	Кнопка
VbOk	1	
VbCancel	2	
VbYes	6	
VbNo	7	
VbRetry	4	

`A=MsgBox ("Ответ печатать?", 3, "Задание № 1")`



### 3. С помощью оператора Print

#### Print список переменных

`Print "d=" & d, "a равно "; a`

#### Format (параметр,"прототип")

**0** – резервирует позицию цифрового разряда, отображает цифру, если у числа, представленного параметром, есть цифра в этой позиции

**#** - аналогичен первому параметру, но не отображаются незначащие нули.

**(точка)** – разделитель целой и дробной части числа

`Print "d="; Format(d,"0.000"), "a равно "; a`

# ЛИНЕЙНЫЙ АЛГОРИТМ

Это основная структура, в которой выполняемые операции следуют одна за другой, образуя простую последовательность.

При программировании структуры можно выделить три части:

- ввод исходных данных;
- преобразование их по заданным формулам;
- вывод полученного результата.

# Структура Следование

## 1 ФОРМАЛИЗАЦИЯ ЗАДАЧИ

### 1.1 Словесное описание

Разработать информационную технологию, позволяющую вычислять расстояние в пространстве от точки, заданной своими координатами до начала координат.

### 1.2 Формализация выходных и входных данных

Выходные данные:

$d$  – расстояние, переменная вещественного типа.

Входные данные:

$x$  – координата точки, константа вещественного типа

$y, z$  – координаты точки, переменные

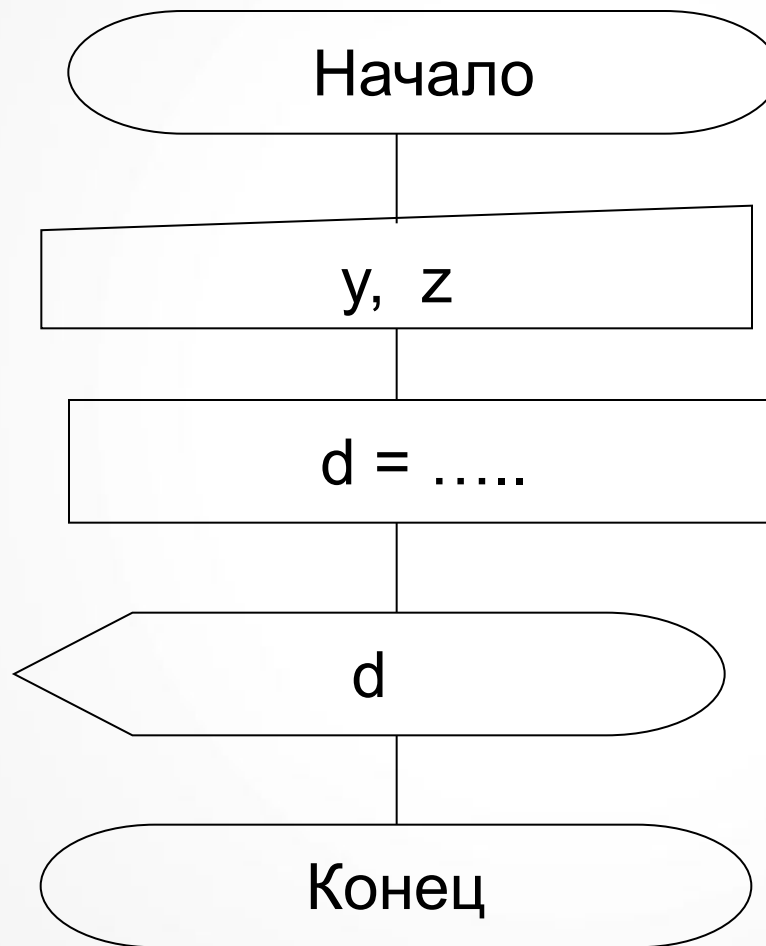
вещественного типа

## 1.3 Определение математических и логических зависимостей

В пространстве расстояние ( $d$ ) от точки, заданной своими координатами ( $x$ ,  $y$ ,  $z$ ) до начала координат определяется по формуле

$$d = \sqrt{x^2 + y^2 + z^2}$$

## 2 СХЕМА АЛГОРИТМА



# Конструктор формы



The image shows a screenshot of a Windows application window titled "Form1". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is light gray and contains the following elements:

- The text "Введите у" (Enter y) is positioned in the upper right area.
- A rectangular text input field is located below the text "Введите у".
- A button with a dashed border and the text "ПУСК" (START) is centered below the first input field.
- The text "d =" is positioned in the lower left area.
- A second rectangular text input field is located below the "d =" text.



# Код приложения

```
Private Sub Command1_Click()  
    Const x As Single = 2.1  
        ' Объявление переменных  
    Dim z As Single, y As Single  
    Dim D As Double  
        ' Ввод значений y и z  
    y = Val(Text1.Text)  
    z = InputBox("Введите значение z")  
        ' Присваивание значения выражения переменной d  
    D = Sqr(x ^ 2 + y ^ 2 + z^2)  
        ' Вывод значений  
    Text2 = D           ' в текстовое поле  
    Print "d равно "; D      ' на форму  
    MsgBox "Значение d равно " & D ' в диалоговое окно  
End Sub
```

Project1



Введите значение z

OK

Cancel

6.3

Project1



Значение d равно 8,07836624109948

OK

Form1

d равно 8,07836624109948

Введите y

4.6

ПУСК

d =

8,0783662410994

## Код приложения

```
Private Sub Form_Click()
```

```
    Const x = 2
```

```
    Dim z As Single, y As Single, d As Single
```

```
    Form1.Caption = "СЛЕДОВАНИЕ"
```

```
    y = Text1
```

```
    z = InputBox("Введите значение z", "Следование")
```

```
    D = Sqr(x ^ 2 + y ^ 2 + z^2)
```

```
    Text2.Text = format(D,"0.0")
```

```
    Print " При x="; x; " и z="; z; Chr(10); "D равно "; _
```

```
        Format (D, "0.00")
```

```
    MsgBox "Значение d равно " & d
```

```
    Text1 = "": Text2 = ""
```

```
    Form1.Cls
```

```
End Sub
```

**СЛЕДОВАНИЕ**

Ввод  $y$ , вывод  $d$

4,6

**Следование**

Введите значение  $z$

OK

Cancel

6,3

**СЛЕДОВАНИЕ**

При  $x=2$  и  $z=6,3$   
 $D$  равно 8,05

Ввод  $y$ , вывод  $d$

4,6

8,1

**Project1**

Значение  $d$  равно 8,05295

OK