Основы языка Visual Basic

1. Типы данных

- Это одно из фундаментальных свойств любого языка программирования.
- В VB определены все стандартные типы данных

Целые типы

- Byte – байт

  1 байт: от 0 до 255

- Integer – целое

  2 байта: от -32768 до 32767

- Long – длинное

  4 байта: от -2^{32} до 2^{32}

  от -2 147 483 648 до 2 147 483 647

Вещественный тип или с плавающей точкой

- Single – одинарная точность 4 байта
  от -3,4E38 до +3,4E38

- Double – двойная точность 8 байт
  от -1,797E308 до +1,797E308

Логический тип

- Boolean – булевский
  2 байта
  0 – False - ложно
  1 – True - истинно

Строковый тип

- String – текстовые и строковые значения длиной от 1 до 65400 символов

Тип data

- Date – 8 байт
  от 01.01.100 по 31.12.9999

Тип Variant

Данные любого типа – более 16 байт

2. Зарезервированные или ключевые слова

- Это такие слова, которые могут применяться в языке VB только в предусмотренном языком смысле и никак иначе.
- Например:

  Integer, Single, If, For, True,…

3. Константа

- Область памяти, имеющая имя.
- Получает своё значение на этапе разработки программы и в процессе её выполнения значение константы изменить нельзя.

- Константа д.б. объявлена:

Const <имя константы>[As< Тип>]= Значение

Например:

Const Pi As Single = 3.1416

4. Переменная

- Область памяти, имеющая имя.
- Получает своё значение на этапе выполнения программы и сохраняет его пока ей не будет присвоено новое значение.
- Переменную необходимо объявить, указав её имя и тип:

Dim <Имя переменной>[As < Тип>]

В одной строке м.б. объявлено несколько переменных:

Dim x As Integer, b As Single, fi As Double

Если тип переменной не объявлен, то ей по-умолчанию присваивается тип Variant

Например:

Dim X, Y as Byte

объявлены переменные:

X типа Variant и

Y типа Byte

На имена в VB накладываются следующие требования

- Используются буквы только латинского алфавита (регистр значения не имеет) и цифры
- В VB имя должно начинаться с буквы
- Имя не должно содержать следующих символов - . ␣ % & ! # @ $

- Имя должно быть уникальным, т.е. не должно совпадать с другими именами или с ключевыми словами;
- Имя не должно содержать > 255 символов;
- Нельзя присваивать переменной значения при её объявлении.

5. Математические операции:

^	возведение в степень	
-	отрицание	
*	умножение	
/	деление	7 / 2 = 3.5
\	целочисленное деление	7 \ 2 = 3
mod	остаток от деления на целое	7 mod 2 = 1
+	сложение	
-	вычитание	

Запись арифметических выражений

1. Выражение записывается в строку
2. Дробная часть числа отделяется точкой
3. Операции выполняются слева направо с учётом приоритетов и круглых скобок
4. Нельзя опускать знак умножения
5. Операция mod с двух сторон ограничивается пробелами - A ␣ mod ␣ B
6. Для переноса выражения на новую строку используются символы «␣_ » в конце строки
7. Числа с множителем 10 в степени представляются в экспоненциальной форме

$5{,}25*10^8$ - 5.25E8

Например:

$\frac{a}{-b} \longrightarrow a / - b$

$ab \longrightarrow a * b$

$\frac{a+b}{cd} \longrightarrow (a + b) / (c * d)$ или $(a+b)/c/d$

6. Оператор присваивания

<Имя переменной> = <Выражение>

- Символ “ = “ понимается не как равенство, а как процесс присвоения значения, полученного в результате вычисления выражения, записанного справа от знака “ = “ переменной записанной слева от этого знака.
- При этом прежнее значение переменной слева от знака присваивания замещается значением вычисленного выражения.

7. Функции в языке VB

- Аргументы записываются после имени функции в круглых скобках и отделяются друг от друга запятыми.
- Функции м.б. математические, строковые, финансовые, даты и др.

Математические функции

Sin(x) - sinx

Cos(x) - cosx

Tan(x) - tg x

Atn(x) - arctg x

Sqr(x) - квадратный корень

Log(x) - натуральный логарифм

Exp(x) - e^x

Abs(x) - | x |, модуль, абсолютное значение

Аргумент тригонометрических функций должен быть представлен в радианной мере!

Вспомним:

Радиан = Градус * Pi / 180

Градус = Радиан * 180 / Pi

Lg x = Log (x) / Log (10)

Int(x) – наибольшее целое число, не превышающее x (без округления).

Cint(x) – ближайшее к x целое число (округл.)

Rnd(x) – случайное число от 0 до 1

Sgn(x) – знак числа (сигнатура)

- 1 – для отрицательного x

+1 – для положительного x

0 – для нуля.

Функции преобразования типов данных

- Функция Val – преобразование строкового значения в числовое. Часто используется для преобразования строкового значения текстового поля или функции InputBox в число, которое затем используется в арифметических операциях.

- Функция Str – осуществляет обратное преобразование числа в строку и служит для вывода в текстовое поле или на панель сообщений функцией MsgBox

Функция Format

Format (<Параметр> , “прототип”)

Например:

Format (X , “# #.# # #”)

Или Format (X , “00.000”)

Значение X будет печататься с двумя цифрами в целой части и тремя цифрами в дробной части.

В первом случае незначащие нули не печатаются.

Функции даты и времени

- Функция Date – возвращает значение текущей даты, которое можно присвоить переменной типа Date.

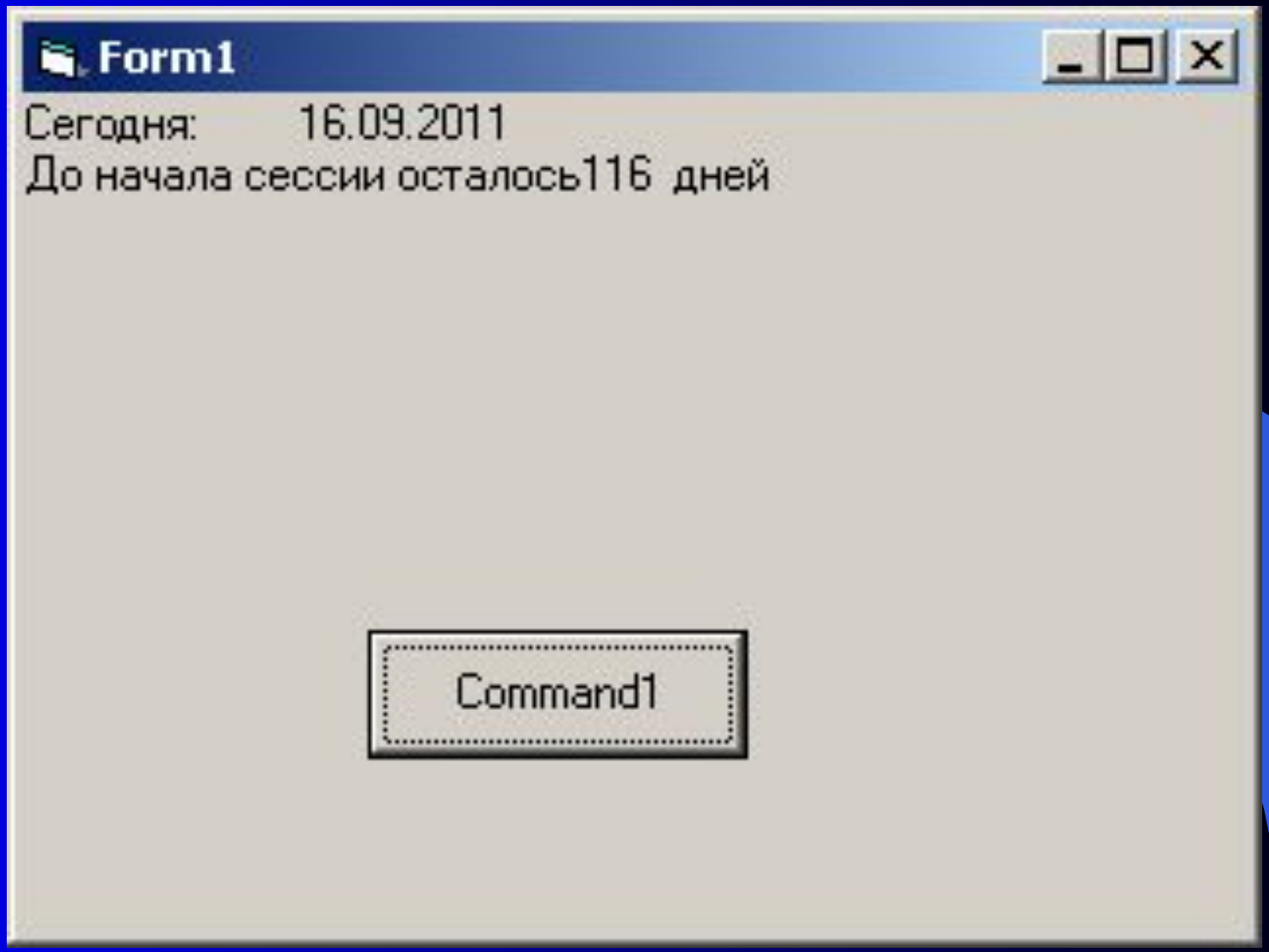
Месяц / Число / Год#

Разностью значений переменных типа Date является число дней между датами.

Программа печатает текущую дату и количество дней до начала зимней сессии:

```
Private Sub Command1_Click()
Dim A As Date, B As Date
A = Date
B = #1/10/2012#
Print "Сегодня:", A
Print "До начала сессии осталось", B-A, "дней"
End Sub
```

В результате имеем :



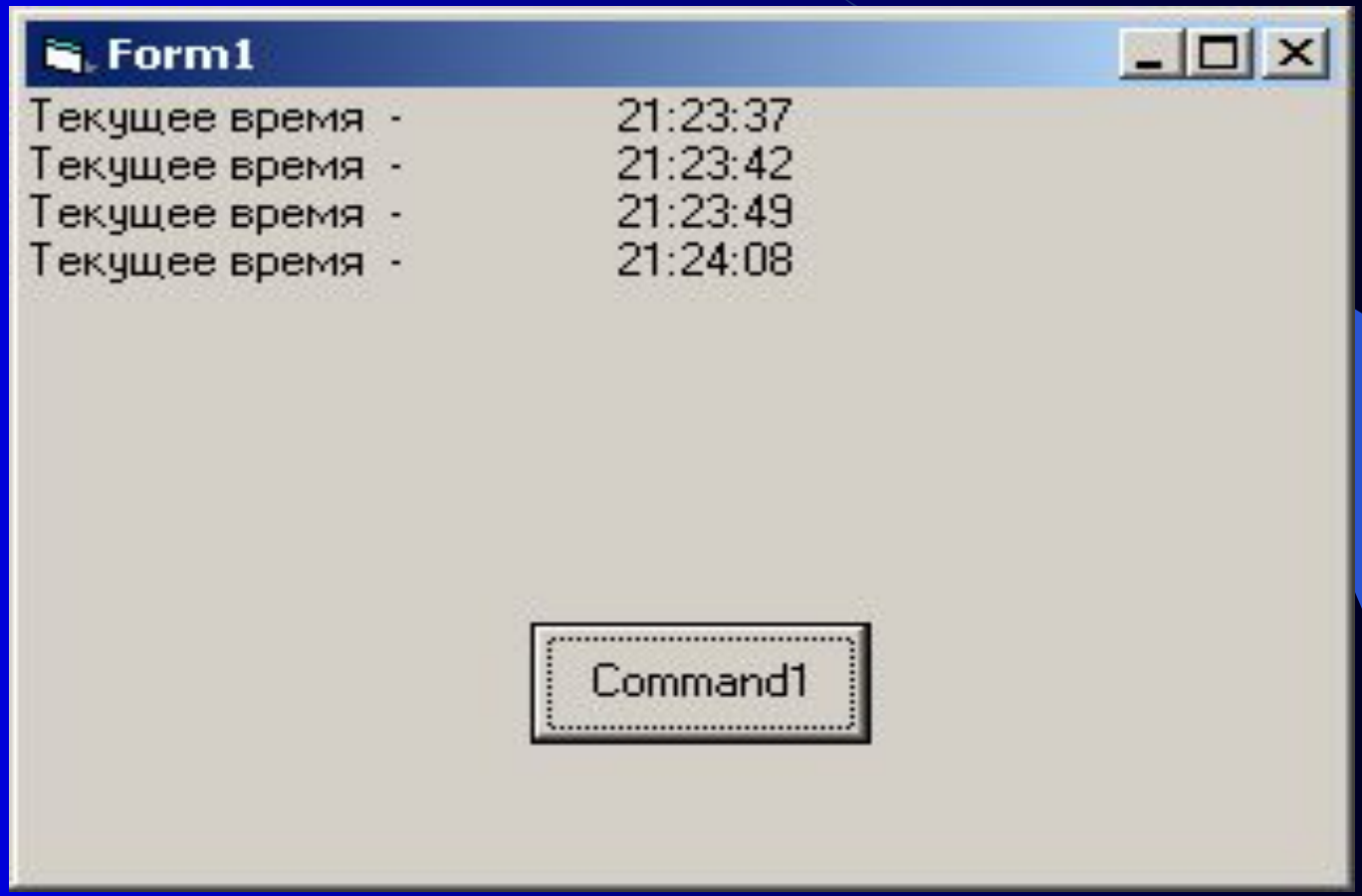
- Функция Time– возвращает значение текущего времени, которое можно присвоить переменной типа String в виде:

Часы : Минуты : Секунды

Программа определяющая текущее время:

```
Private Sub Command1_Click()
Dim A As String
A = Time
Print "Текущее время - ", A
End Sub
```

Каждая новая строчка печатается после нажатия кнопки Command1



8. Организация ввода данных

Ввод данных
можно осуществлять с помощью

- текстового поля
- с помощью функции *InputBox()*

Ввод с помощью текстового поля

Мы рассмотрели на примере вычисления длины окружности.

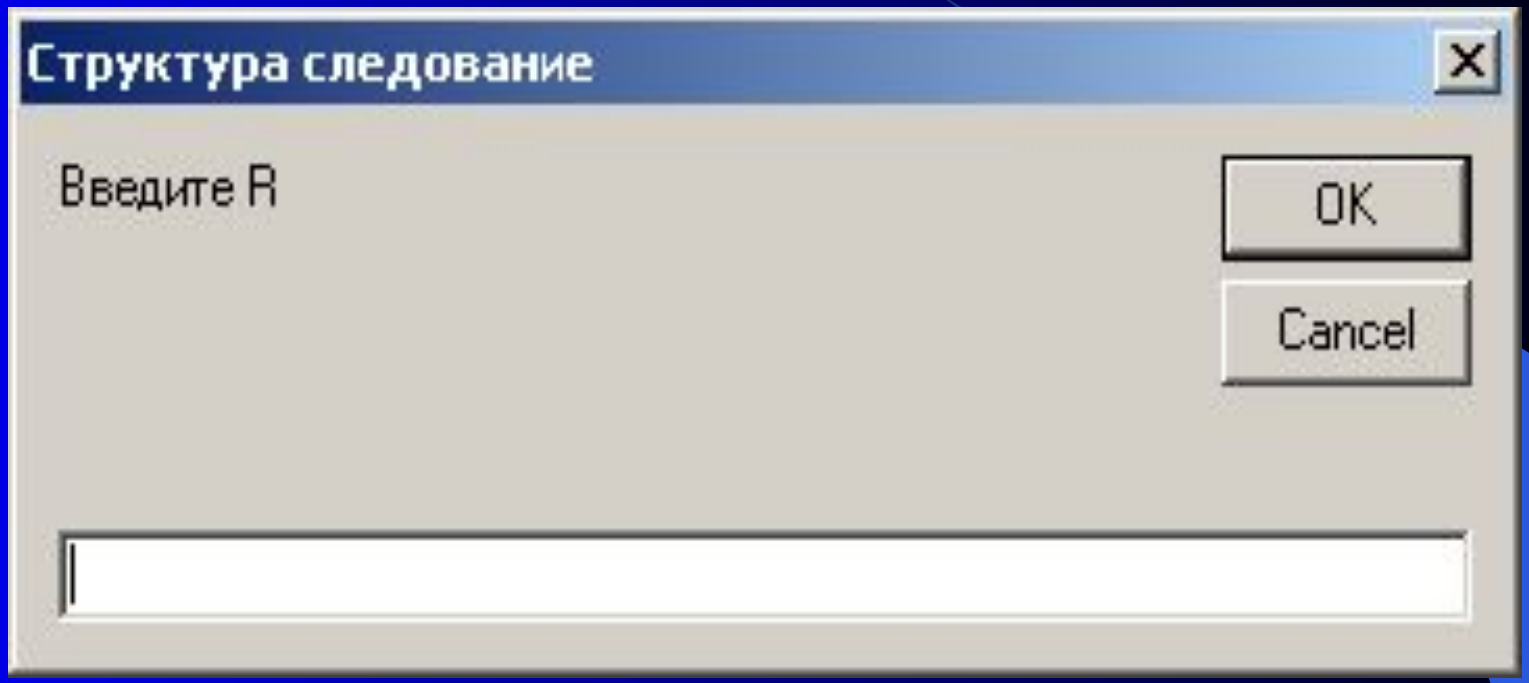
- На форме разместили текстовое поле Text1
- В тексте программы записали оператор
  R = Val(Text1.text)

Функция ввода – *InputBox()*

- Эта функция инициирует создание диалогового окна с текстовым полем для ввода данных
- Имеет три аргумента

InputBox(<Приглашение>,<Заголовок>,[По умолчанию])

R=InputBox("Введите R", "Структура следование")



- В строке заголовка печатается значение аргумента «Заголовок».
- На самой панели печатается значение аргумента «Приглашение».
- В текстовом поле печатается значение аргумента «По умолчанию».
  Если этот аргумент отсутствует, то отсутствует и содержимое текстового окна
- Введенная пользователем в текстовом поле строка становится значением функции

9. Организация вывода данных

С помощью :

- текстового поля
- метода Print
- оператора MsgBox
- окна списка ListBox

Вывод с помощью текстового поля

См. пример вычисления длины окружности:

- на форме размещено текстовое поле Text2.text
- в программе записан оператор Text2.text = Str(L)

Метод Print – выводит результаты на форму

- Если выводимые значения в списке разделяются символом « ; », то они печатаются через один пробел.
- Если символом « , », то каждое следующее значение печатается через 14 пробелов.
- Также выводимые значения могут разделяться символом & (конкатенация). В этом случае они печатаются слитно

- Функция Spc(число) печатает заданное число пробелов.
- Функции

  CurrentX = число

  CurrentY = число

  указывают координаты начала печати на форме.

NB!

- Каждый оператор Print печатает с новой строки,

  но если оператор Print заканчивается запятой или точкой с запятой, то следующий оператор Print начнет печатать результаты в той же строке, в которой напечатал результаты предыдущий оператор Print

MsgBox – панель сообщений в режиме оператора

- используется для вывода данных на специальную панель сообщений

MsgBox<Сообщение>,[<Код1> + <Код2>],[Заголовок]

Сообщение – текст, отображаемый в диалоговом окне,

Код1 – определяет вид пиктограммы, которая помещается на панель сообщений,

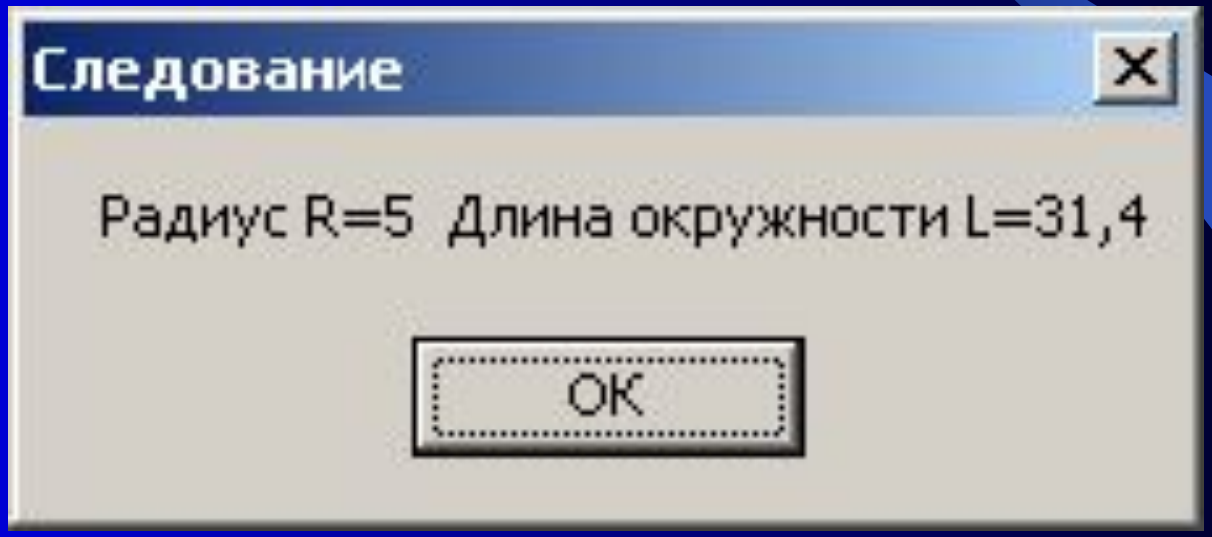
Код2 – определяет набор кнопок размещаемых на панели,

Заголовок – печатается в строке заголовка панели.

Например:

MsgBox «Радиус R=» & R & « Длина окружности L=» & L, , «Следование»

NB! Отсутствующий второй аргумент заменяется запятой



После нажатия кнопки ОК выполнение программы продолжается.

- Внешний вид панели сообщений можно менять используя различные значения Код1 и Код2

С помощью одного числа, являющегося суммой Код1+Код2 можно одновременно устанавливать пиктограмму и комбинацию кнопок на панели сообщений

Код1	Пиктограмма
16	
32	?
48	!
64	i

Код2	Набор кнопок
0	ОК
1	ОК, Отмена
2	Стоп, Повтор, Пропустить
3	Да, Нет, Отмена
4	Да, Нет
5	Повтор, Отмена

MsgBox в режиме функции

- Если аргументы записывать в скобках, то функция MsgBox получает определенное значение, которое м.б. присвоено какой-либо переменной.

<Имя переменной> =
MsgBox(<Сообщение>,[<Код1>+<Код2>],[Заголовок])

Значение функции MsgBox в зависимости от нажатой кнопки:

OK……..1
Отмена……..2
Стоп……..3
Повтор……..4
Пропустить……..5
Да……..6
Нет……..7

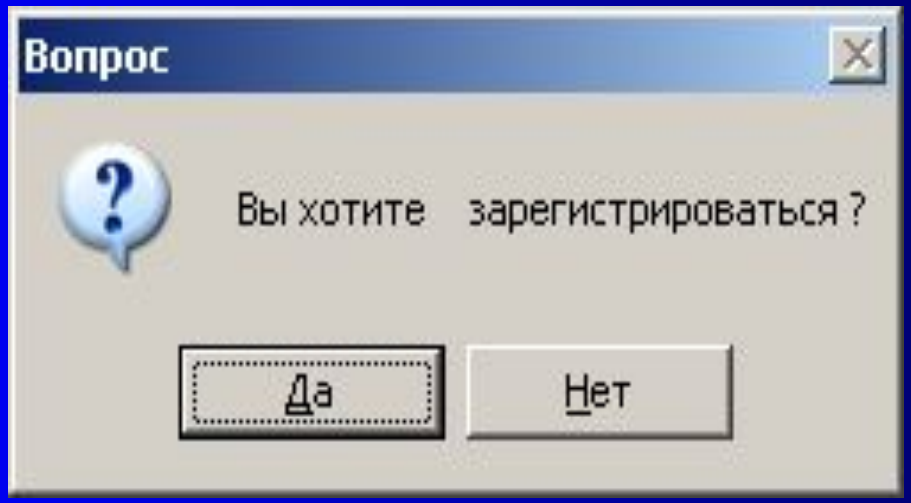
Рассмотрим процедуру регистрации пользователя:

```
Private Sub Command1_Click()
    Dim A As Byte
    A = MsgBox("Вы хотите
    зарегистрироваться ?", 36, "Вопрос")
    MsgBox   A
End Sub
```

Функция MsgBox выводит панель сообщений.

Щелчок по кнопке определяет значение, которое будет присвоено переменной A.

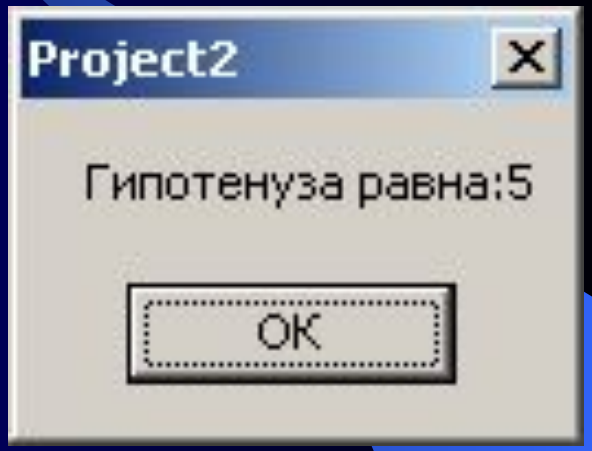
Оператор MsgBox печатает это значение на панели сообщений.



Рассмотрим пример вычисления гипотенузы прямоугольного треугольника по двум его катетам:

```
Private Sub Command1_Click()
Dim A As Single, B As Single, C As Single
A = InputBox("Первый катет:", "Катет A")
B = InputBox("Второй катет:", "Катет B")
C = Sqr(A ^ 2 + B ^ 2)
MsgBox " Гипотенуза равна:  " & C
End Sub
```


Катет А
Первый катет:
OK
Cancel
3


Project2
Гипотенуза равна:5
OK


Катет В
Второй катет:
OK
Cancel
4

10. Операции отношения и логические операции

Операции отношения : < , > , <= , >= , <> ,=

их результатом всегда является логическое (булево) значение, выражающее истинность некоторого отношения между данными (операндами)

Операция A > B может иметь значения:

True – истинно

или

False – ложно

- Логические операции – выполняются над логическими значениями или выражениями, результатом которых являются логические значения. В результате получаются также логические значения:

Not – логическое **Не** - отрицание
And – логическое **И** - конъюнкция
Or – логическое **Или** - дизъюнкция

A	B	Not A	A And B	A Or B	A Xor B	A Eqv B	A Imp B
T	T	F	T	T	F	T	T
T	F	F	F	T	T	F	F
F	T	T	F	T	T	F	T
F	F	T	F	F	F	T	T