



ПЕРЕМЕННЫЕ, АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ ЯЗЫКА

лекция 3

ПЕРЕМЕННЫЕ

- ▣ **Переменной** называется та величина, значение которой может изменяться по ходу выполнения программы. Значениями переменных могут быть свойства объекта, который мы размещаем на форме: цвет, размеры, текст и все, что можно видеть в окне Properties, а также числовые данные, меняющиеся в процессе математических расчетов.
- ▣ **Переменная**- ячейка памяти в которой хранятся данные



ОПЕРАТОР ПРИСВАИВАНИЯ

Используется для задания переменным различных значений.

- ▣ Переменным могут присваиваться числовые, текстовые значения, значения свойств объектов, других переменных. Переменные могут также представлять объекты, например формы или элементы управления.



ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ

- В Visual Basic .NET необходимо явно объявлять переменные перед их использованием.
- Объявить переменную - значит создать ее. Для этого переменной надо присвоить имя, указать тип данных, которые будут использоваться в качестве ее значений, а также написать ключевые слова.
- *Dim Имя As тип данных*
- *Private Имя As тип данных*
- *Public Имя As тип данных*



ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ

- **Имя** может состоять из букв, цифр, знаков подчеркивания, но начинаться должно только с буквы. Недопустимо использование пробелов, знаков препинания, а также слов, используемых Visual Basic .NET (New, Handles, End, Call, Sub, буква e латинского регистра и так далее).
- Размер имени может колебаться от одного символа до нескольких тысяч
- Если известно значение переменной на этапе ее описания:

```
Dim strName As String = "Bob Benzel"
```



НОВЫЕ СОКРАЩЕННЫЕ ОПЕРАТОРЫ

Операция	Длинная запись	Короткая запись
Сложение (+)	$X = X + 6$	$X += 6$
Вычитание (-)	$X = X - 6$	$X -= 6$
Умножение (*)	$X = X * 6$	$X *= 6$
Деление (/)	$X = X / 6$	$X /= 6$
Целочисленное деление (\)	$X = X \setminus 6$	$X \setminus = 6$
Возведение в степень (^)	$X = X ^ 6$	$X ^= 6$
Конкатенация строк (&)	$X = X \& \text{"ABC"}$	$X \&= \text{"ABC"}$

ТИПЫ ДАННЫХ

Тип	Определение	Диапазон	Размер	Префикс
Integer	Целые числа	от -2 млрд до +2 млрд	4 Б	int
Long	Длинное целое	Все целые	8 Б	lng
Single	Дробные числа одинарной точности	-3,4E38:: -1,4E-45; 1,4E-45:: 3,4 E38	4 Б	sng
Double	Дробные числа двойной точности	-1,79E308:: -4,94E-324; 4,94E-324:: 1,79E308	8 Б	dbl
Byte	Целые	0-255	1 Б	byt
Char	Символьный	0-65535	2 Б	chr



ТИПЫ ДАННЫХ

Тип	Определение	Диапазон	Размер	Префикс
Short	Короткое целое	от -32 768 до 32 767	2 Б	sho
String	Строка	0 -2 000000	4 Б	str
Date	Даты	До 9999 г.		dte
Boolean	Логическое	И, Л	2 Б	bin
Decimal	Любые		16 Б	dec



ТИПЫ ДАННЫХ

Типы данных, определяемые пользователем:

```
Structure Employee  
Dim Name As String  
Dim DateOfBirth As Date  
Dim HireDate As Date  
End Structure
```

```
Dim ProductManager As Employee  
ProductManager.Name = "Эрик Коди"
```



ОБЛАСТЬ ВИДИМОСТИ ПЕРЕМЕННЫХ

Областью видимости переменных может быть блок, процедура, модуль, вся программа.

- видимость в пределах блока:

Dim имя переменной *As* тип данных

- видимость в пределах модуля:

Private имя переменной *As* тип данных

- видит вся программа – глобальные:

Public имя переменной *As* тип данных

пишется вначале файла



КЛАССЫ

- ▣ **Class** класс - это основа, на которой строится все программирование в Visual Basic .Net.
- ▣ Класс объединяет в себе методы, свойства и события.
- ▣ Любой объект, расположенный на форме, можно представить как экземпляр класса.
- ▣ А класс представляет собой инкапсуляцию (объединение) данных этого объекта и методов, которые обрабатывают эти данные.
- ▣ *View > Object Browser* отображает библиотеки классов



КЛАСС МATH

Метод	Описание
Abs(a)	Абсолютное значение числа a
Acos(a) Asin (a) Atan (a)	Арккосинус, Арксинус, Арктангенс числа a
Cos (a)	Косинус числа a
E(a)	Основание натурального логарифма
Exp (a)	Экспонента от числа a
Floor (a)	Наибольшее целое, меньшее или равное a
lg (a)	Логарифм числа a
Log10(a)	Десятичный логарифм указанного числа a
Max(a,b)	Максимальное из двух чисел a и b
Min(a,b)	Минимальное из двух чисел a и b
Pow(a, b)	Возведение числа a в степень b
PI	Число PI
Rnd ()	Генерирует случайное число от 0 до 1
Sin (a)	Синус числа a
Sqrt(a)	Корень квадратный числа a
Tan (a)	Тангенс числа a



КЛАСС MATH

□ Вызов класса:

Imports System.Math

Пример:

Math.Sqrt (64)

Math.Pow (2,3) = 2³

Операторы	Порядок вычисления
()	Первыми всегда вычисляются значения в круглых скобках.
^	Возведение числа в степень.
-	Создание отрицательного числа (смена знака)
* /	умножение и деление.
\	целочисленное деление.
Mod	остаток от деления.
+ -	Последние - это сложение и вычитание.



ПРЕОБРАЗОВАНИЯ РАЗНОТИПНЫХ ЗНАЧЕНИЙ

- В VB.NET предусмотрен режим жесткой проверки типов: **OPTION Strict** (Build/Project Properties/strict Type Checking)
- Функции явного преобразования типов:
 - CBool, CByte, CInt, CLng, CSng, CDate, CDb1, CDec, CStr, CChar



СТРОКОВЫЕ ФУНКЦИИ КЛАССА STRING

Функция	Описание
ASC	Возвращает код первого символа в строке
Chr	Преобразует число в символ
Filter(S, S1)	Получает строку S и искомую строку S1; возвращает одномерный массив всех эл-ов из S1
GetChar (S, N)	Возвращает n-ый символ строки S в формате char
InStr (S, S1)	Возвращает позицию первого вхождения S1 в S
InStrRev(S, S1)	Возвращает позицию последнего вхождения S1 в S
Join	Строит большую строку из меньших строк
Lcase/Ucase	Преобразует строку к нижнему/верхнему регистру
Left/Right (S,N)	Находит и выделяет n символов с лева/права строки
Len	Возвращает длину строки
Ltrim/Rtrim/Trim	Удаляет пробелы в начале/конце/начале и конце строки
Mid (S, N, K)	Выделяет подстроку в строке



СТРОКОВЫЕ ФУНКЦИИ КЛАССА STRING

Функция	Описание
Replace	Заменяет одно или более вхождений одной строки в другой строке
Space	Генерирует строку заданной длины, состоящую из пробелов
Split	Позволяет разбивать строку по заданным разделителям (например, пробелам)
Str	Возвращает строковое представление числа
StrComp	Альтернативный способ сравнения строк
StrConv	Преобразует строку из одной формы в другую (например, с изменением регистра)
String	Создает строку, состоящую из многократно повторяющегося символа



СТРОКОВЫЕ МЕТОДЫ .NET FRAMEWORK

Метод	Описание
Chars	Возвращает символ, находящийся в заданной позиции
Compare	Сравнивает две строки
Copy	Копирует строку
CopyTo	Копирует N символов, начиная с K-го, в M-ый элемент массива символов
Empty	Пустая строка
EndsWith(S, S1)	Завершается ли S подстрокой S1
IndexOf	Возвращает индекс первого вхождения S1 в S
Insert(S, S1, n)	Вставка подстроки S1 в строку S с N-ой позиции
Length	Количество символов в строке
Replace (S, S1,S2)	Замена одной подстроки на другую



СТРОКОВЫЕ МЕТОДЫ .NET FRAMEWORK

Join	Объединяет массив строк с заданным разделителем
LastIndexOf	Возвращает индекс последнего вхождения заданного символа или подстроки в строке
PadLeft PadRight	Выравнивает символы строки по правому/ левому краю. Строка дополняется слева пробелами или другими символами до заданной длины
Remove	Удаляет из строки заданное количество символов, начиная с заданной позиции
Split	Разбивает строку, превращая ее в массив подстрок
StartsWith	Проверяет, начинается ли заданная строка определенной последовательностью символов
Substring	Возвращает подстроку, начинающуюся с заданной позиции
ToCharArray	Копирует символы строки в символьный массив
ToLower/ ToUpper	Возвращает копию строки, преобразованную к нижнему/верхнему регистру
Trim/ TrimEnd TrimStart	Удаляет пробелы или все символы из набора, заданного в виде массива символов Unicode, в начале и/или конце строки



СТРОКОВЫЕ ФУНКЦИИ

- Конкатенация строк: «мой» & « дядя»
- Класс **String** при работе со строками создает копию строки и потом ее преобразует. Поэтому для его работы требуется больше ресурсов.
- Класс **StringBuilder** используется, когда необходимо ввести постоянные изменения в строку
- При программировании для .NET рекомендуется использовать методы и свойства класса **String**:

```
dim r as string = "Mississippi"  
Console.WriteLine(r.Replace("i", "o"))
```



КЛАССЫ КОЛЛЕКЦИЙ В .NET

- ArrayList – динамический массив
- BitArray- используется для поразрядных операций с битами
- DictionaryBase- базовый класс для ассоциативных массивов (ключ/значение)
- Hashtable- коллекция пар «ключ/значение» упорядоченная по хэш-кодам ключей
- Queue- очередь (FIFO)
- Stack – стек (LIFO)



ОПЕРАТОРЫ ЯЗЫКА

□ Условные команды

- If <условие> then

<команды>

End If

- If <условие> then

<команды>

Else

<команды>

End If

- If <условие> then

<команды>

Else If <условие> then

<команды>

End If



ОПЕРАТОРЫ ЯЗЫКА

□ Циклы

- For i=0 to N

<команды>

next i

- For i=0 to N

For J=0 to M

<команды>

next J, I

Exit For- принудительный выход из цикла (прерывание)



ОПЕРАТОРЫ ЯЗЫКА

□ Циклы с условием

- DO

<команды>

LOOP Until <условие>

- DO

<команды>

LOOP While <условие>

- DO Until <условие>

<команды>

LOOP

- DO While <условие>

<команды>

LOOP



ОПЕРАТОРЫ ЯЗЫКА

□ Выбор

Select Case <проверяемая переменная>

Case <значение /условие/ диапазон значений>

<команда выполняемая при значении «Истина»>

Case <значение /условие/ диапазон значений>

<команда выполняемая при значении «Истина»>

Case Else

<команда выполняемая при значении «Ложь»>

End Select

Возможно использование логических операторов: AND,
OR, NOT



МАССИВЫ

- Объявление массивов: Dim, Private, Public

Dim a (2) As String

a (0) = "Label1"

a (1) = "Label2"

a (2) = "Label3"

Двумерный массив- Матрица чисел:

Dim b(2,3) as byte

- Динамический массив: **dim a() as integer**

ReDim a(5)



МАССИВЫ

- Заполнение массива:

```
Dim a(10) as byte
```

```
For i=0 to 9
```

```
a(i)= rnd * 100
```

```
Next i
```

- Двумерные массивы

```
Dim s (6,4) as byte
```

```
For i=0 to 5
```

```
For j=0 to 4
```

```
S(i, j)= rnd * 100
```

```
Next j, i
```



ПРИМЕР ВЫВОДА ДВУМЕРНОГО МАССИВА

- T1.Text = ""
- Dim a(2, 2) As Byte
- For i = 0 To 1
- For j = 0 To 1
- a(i, j) = CByte(Rnd() * 100)
- T1.Text = T1.Text & a(i, j) & " "
- Next j
- T1.Text = T1.Text + vbCrLf
- Next i



ПРЕОБРАЗОВАНИЕ СТРОКИ В МАССИВ

- `Dim s() As String`
- `Dim a As String`
- `a = T2.Text`
- `s = Split(a)`
- `For i = 0 To UBound(s)`
- `T1.Text = T1.Text + s(i) + _`
- `vbCrLf`
- `Next`

