



# Основы программирования на Visual Basic.NET

## Типы данных и переменные

Последнее обновление: 30.10.2015



Все данные, используемые в Visual Basic.NET, описываются целой системой типов данных. Эта система определяет следующие примитивные типы данных:

- **Boolean**: представляет логическое значение. Может принимать True или False. Представлен системным типом `System.Boolean`
- **Byte**: представляет 8-ми битное число от 0 до 255 и занимает 1 байт. Представлен системным типом `System.Byte`
- **Short**: представляет число от -32768 до 32767 и занимает 2 байта. Представлен системным типом `System.Int16`
- **Integer**: представляет целое число от -2 147 483 648 до 2 147 483 647 и занимает 4 байта. Представлен системным типом `System.Int32`
- **Long**: представляет целое число от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807 и занимает 8 байт. Представлен системным типом `System.Int64`
- **SByte**: представляет значение от -128 до 127 и занимает 1 байт. Представлен системным типом `System.SByte`
- **UShort**: представляет целое число от 0 до 65 535 и занимает 2 байта. Представлен системным типом `System.UInt16`
- **UInteger**: представляет целое число от 0 до 4 294 967 295 и занимает 4 байта. Представлен системным типом `System.UInt32`
- **ULong**: представляет целое число от 0 до 18 446 744 073 709 551 615 и занимает 8 байт. Представлен системным типом `System.UInt64`

- **Single:** представляет число с плавающей точкой и может принимать следующие значения: для отрицательных чисел от  $-3,4028235E+38$  до  $-1,401298E-45$ ; для положительных чисел от  $1,401298E-45$  до  $3,4028235E+38$ . Представлен системным типом `System.Single`
- **Double:** представляет числа с плавающей запятой двойной точности. Может принимать следующие значения: для отрицательных чисел от  $-1,79769313486231570E+308$  до  $-4,94065645841246544E-324$ ; для положительных чисел от  $4,94065645841246544E-324$  до  $1,79769313486231570E+308$ . В памяти занимает 8 байт. Представлен системным типом `System.Double`
- **Decimal:** хранит десятичное число с фиксированной запятой. Если употребляется без десятичной запятой, имеет значение от 0 до  $\pm 79\,228\,162\,514\,264\,337\,593\,543\,950\,335$ ; если с запятой, то от 0 до  $\pm 7,9228162514264337593543950335$  с 28 разрядами после запятой и занимает 16 байт. Представлен системным типом `System.Decimal`
- **Date:** представляет дату от 0:00:00 1 января 0001 года до 23:59:59 31 декабря 9999 года и занимает 8 байт. Представлен системным типом `System.DateTime`
- **Char:** хранит одиночный символ в кодировке Unicode и занимает 2 байта. Представлен системным типом `System.Char`
- **String:** хранит набор символов Unicode. Представлен системным типом `System.String`
- **Object:** может хранить значение любого типа данных и занимает 4 байта на 32-разрядной платформе и 8 байт на 64-разрядной платформе. Представлен системным типом `System.Object`, который является базовым для всех других типов и классов .NET.

## Объявление переменных

Для объявления переменных мы используем ключевое слово `Dim`, либо один из модификаторов доступа - `Private`, `Friend`, `Protected`, `Public` или `Static`. О модификаторах доступа мы поговорим чуть попозже, а пока ограничимся употреблением оператора `Dim`. Далее мы должны указать имя переменной. Требования к имени переменной следующие:

- имя должно содержать не более 255 символов
- имя может содержать любые буквы, цифры и символ подчеркивания, при этом первый символ в имени должен быть буквой или символом подчеркивания

- в имени должны отсутствовать пробелы и знаки пунктуации
- имя не должно быть ключевым словом

После имени переменной указывается ключевое слово `As` и затем идет тип данных переменной. После указания типа данных мы можем указать также и значение переменной с помощью знака равно. Ниже даны примеры объявления переменных:

```
1 Dim x As Integer
2 Dim y As Double = 4.0
3 Dim s As String = "Hello"
```

Это были примеры явного указания переменных, то есть мы явно указывали их тип данных. Однако возможен и другой способ, когда компилятор выводит тип данных из ее значения, то есть неявный способ определения переменных. Поэтому уместны и следующие объявления:

```
1 Dim x = 6
2 Dim y = 4.0
3 Dim s = "Hello"
```

Если однажды объявив переменную, мы можем многократно изменять ее значение, то значение констант мы изменять не можем и можем указать его лишь однажды. Объявление константы похоже на объявление переменной, только вместо слова `Dim` используется ключевое слово `Const`:

```
1 Const x As Integer = 6
2 Const y = 4.0
```

## Использование системных типов

При перечислении всех базовых типов данных, упоминалось, что каждому примитивному типу данных соответствует какой-то системный тип. Потому что название встроенного типа по сути представляет собой сокращенное обозначение системного типа. Например, следующие переменные будут эквивалентны по типу:

```
1 Dim a As Integer = 5
2 Dim d As System.Int32 = 5
```

[Назад](#) [Содержание](#) [Вперед](#)



ТАКЖЕ НА METANIT.COM

ListView

2 месяца назад · 1 коммент...

ListView в JavaFX, создание списков, получение выбранных в ...

Подключение к SQLite

5 месяцев назад · 1 коммен...

Библиотека sqlite3, подключение к базе данных SQLite в ...

Встроенные компоненты ввода

5 месяцев назад · 1 коммен...

Встроенные компоненты ввода Blazor из пространства имен ...

Взаим

5 меся

Взаим Python языке

3 Комментариев

1 Войти ▾

G

Присоединиться к обсуждению...

войти с помощьюили через DISQUS ?

Имя

♡

Поделиться

ЛучшиеНовыеСтарые

Anton Zaicev

6 лет назад

Single, было бы круто подписать сколько занимает байт))

10 Ответить • Поделиться ›

A

Alexandr

8 лет назад

"SByte: редставляет ..." - опечаточка ;)

10 Ответить • Поделиться ›

Metanit

Модератор

8 лет назад

поправил

→ Alexandr

00 Ответить • Поделиться ›

Подписаться

О защите персональных данных

**YooMoney:**

410011174743222

**Qiwi:**[qiwi.com/n/METANIT](https://qiwi.com/n/METANIT)

Перевод на карту

**Номер карты:**

4048415020898850

[Вконтакте](#) | [Телеграм](#) | [Twitter](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2023. Все права защищены.