Тема 2: Типы данных и переменные

- Как и во многих языках программирования, в C# есть своя система типов данных, которая используется для создания переменных. Она представлена следующими типами:
- **bool**: хранит значение true или false. Представлен системным типом System.Boolean
- **byte**: хранит целое число от 0 до 255 и занимает 1 байт. Представлен системным типом System.Byte
- **sbyte**: хранит целое число от -128 до 127 и занимает 1 байт. Представлен системным типом System.SByte
- **short**: хранит целое число от -32768 до 32767 и занимает 2 байта. Представлен системным типом System.Int16

- **ushort**: хранит целое число от 0 до 65535 и занимает 2 байта. Представлен системным типом System.UInt16
- **int**: хранит целое число от -2147483648 до 2147483647 и занимает 4 байта. Представлен системным типом System.Int32
- **uint**: хранит целое число от 0 до 4294967295 и занимает 4 байта. Представлен системным типом System.UInt32
- **long**: хранит 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
- **ulong**: хранит целое число от 0 до 18 446 744 073 709 551 615 и занимает 8 байт. Представлен системным типом System.UInt64
- **float**: хранит число с плавающей точкой от $-3.4*10^{38}$ до $3.4*10^{38}$ и занимает 4 байта. Представлен системным типом System.Single

- **double**: хранит число с плавающей точкой от $\pm 5.0*10^{-324}$ до $\pm 1.7*10^{308}$ и занимает 8 байта. Представлен системным типом System.Double
- **decimal**: хранит десятичное дробное число. Если употребляется без десятичной запятой, имеет значение от 0 до +/–79 228 162 514 264 337 593 543 950 335; если с запятой, то от 0 до +/–7,9228162514264337593543950335 с 28 разрядами после запятой и занимает 16 байт. Представлен системным типом System.Decimal
- **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

- Общий способ объявления переменных следующий: тип_данных название_переменной. Например, int x;. В этом выражении мы объявляем переменную x типа int. То есть x будет хранить некоторое число не больше 4 байт.

В качестве имени переменной может выступать любое произвольное название, которое удовлетворяет следующим требованиям:

- имя должно содержать не более 255 символов
- имя может содержать любые цифры, буквы и символ подчеркивания, при этом первый символ в имени должен быть буквой или символом подчеркивания
- в имени не должно быть знаков пунктуации и пробелов
- имя не может быть ключевым словом языка C#.

Варианты объявления переменных

- bool isEnabled = true;
- int x;
- double y=3.0;
- string hello="Hello World";
- char c='s';
- int a=4;
- int z=a+5;

Использование суффиксов

- При присвоении значений надо иметь в виду следующую тонкость. При присвоении переменным типа float и decimal чисел с плавающей точкой, Visual Studio рассматривает все эти числа как значения типа double. И чтобы конкретизировать, что данное значение должно рассматриваться как float, нам надо использовать суффиксы (f и m соответственно для float и decimal):
- float b = 30.6f;
- decimal d = 334.8m;

Использование системных типов

- Выше при перечислении всех базовых типов данных для каждого упоминался системный тип. Потому что название встроенного типа по сути представляет собой сокращенное обозначение системного типа. Например, следующие переменные будут эквивалентны по типу:
- int a = 4;
- System.Int32 b = 4;

Неявная типизация

- Ранее мы явным образом указывали тип переменных, например, int x;. И компилятор при запуске уже знал, что x хранит целочисленное значение.
- Однако мы можем использовать и модель неявной типизации:
- var stroka = "Hell to World";
- var c = 20;
-
- Console.WriteLine(c.GetType().ToString());
- Console.WriteLine(stroka.GetType().ToString();

- Для неявной типизации вместо названия типа данных используется ключевое слово var. Затем уже при компиляции компилятор сам выводит тип данных исходя из присвоенного значения. В примере выше использовалось выражение Console.WriteLine(c.GetType().ToString());, которое позволяет нам узнать выведенный тип переменной c. Так как по умолчанию все целочисленные значения рассматриваются как значения типа int, то поэтому в итоге переменная c будет иметь тип int или System.Int32

- Эти переменные подобны обычным, однако они имеют некоторые ограничения.
- Во-первых, мы не можем сначала объявить неявно типизируемую переменную, а затем инициализировать:
- // этот код работает
- int a;
- a = 20;
-
- // этот код не работает
- var c;
- c= 20;

- Во-вторых, мы не можем указать в качестве значения неявно типизируемой переменной null:
- // этот код не работает
- var c=null;