В C# имеются две общие категории встроенных типов данных: *типы значений* и *ссылочные*

*типы.* Они отличаются по содержимому переменной. Если переменная относится

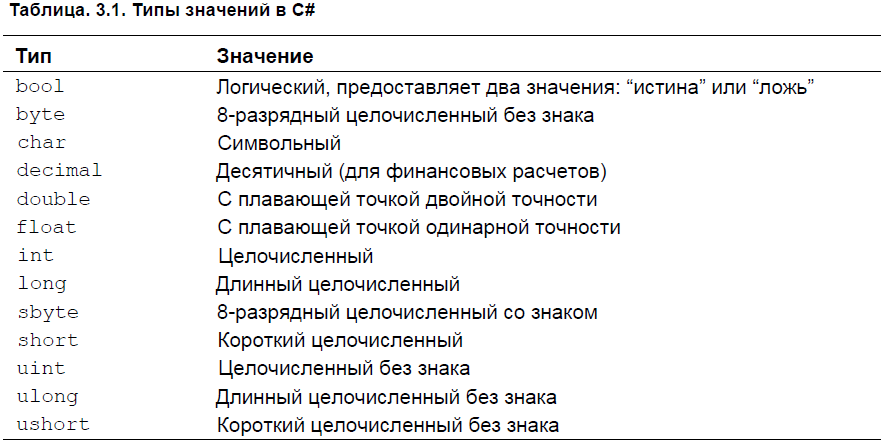
к типу значения, то она содержит само значение, например, 3,1416 или 212.

А если переменная относится к ссылочному типу, то она содержит ссылку на значение.

В основу языка C# положены 13 типов значений, перечисленных в табл. 3.1. Все они

называются *простыми типами,* поскольку состоят из единственного значения. (Иными

словами, они не состоят из двух или более значений.)



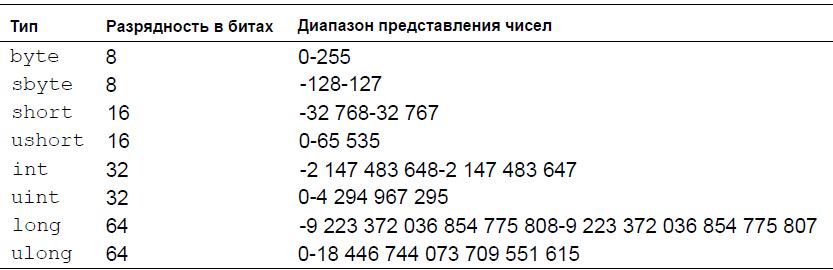
В C# определены девять целочисленных типов: char, ***byte***, ***sbyte***, ***short***, ***ushort***,

***int***, ***uint***, long и ***ulong***. Но тип ***char*** применяется, главным образом, для представления

символов и поэтому рассматривается далее в этой главе. Остальные восемь

целочисленных типов предназначены для числовых расчетов. Ниже представлены их

диапазон представления чисел и разрядность в битах. (***на картинке между числами тире(-) а не минус***)



Вместо ***unsigned int***, просто ***uint***.

class Inches

{

static void Main()

{

long inches, miles;

miles = 930000000;

inches = miles \* 5280 \* 12;

Console.WriteLine("Distance to the Sun: " + inches + " inches.");

}

}

Byte=========================================================================

Самыми мелкими целочисленными типами являются ***byte*** и ***sbyte***. Тип ***byte***

представляет целые значения без знака в пределах от ***0*** до ***255***. Переменные типа ***byte***

особенно удобны для обработки исходных двоичных данных, например байтового потока,

поступающего от некоторого устройства. А для представления мелких целых значений

со знаком служит тип ***sbyte***.

Вероятно, самым интересным среди всех числовых типов данных в ***C#*** является тип

***decimal***, который предназначен для применения в финансовых расчетах. Этот тип

имеет разрядность 128 бит для представления числовых значений в пределах от ***1Е-28***

до ***7,9Е+28***.

class UseDecimal

{

static void Main()

{

decimal price;

decimal discount;

decimal discounted\_price;

price = 19.95m;

discount = 0.15m;

discounted\_price = price - (price \* discount);

Console.WriteLine("Price w discount: " + discounted\_price);

}

}

Обратите внимание на то, что значения констант типа ***decimal*** в приведенном

выше примере программы указываются с суффиксом ***m***. Дело в том, что без суффикса

***m*** эти значения интерпретировались бы как стандартные константы с плавающей точкой,

которые несовместимы с типом данных ***decimal***.

Короче говоря, этот тип предназначен для очень точных исчислений, потому что отображают очень большие числа после запятой или вообще.

Char=========================================================================

В ***C#*** символы представлены не 8-разрядным кодом, как во многих других языках

программирования, например, ***C++***, а 16-разрядным кодом, который называется ***уникодом***

(***Unicode***). Для того чтобы присвоить значение символьной переменной, достаточно заключить

это значение (т.е. символ) в одинарные кавычки.

bool=========================================================================

Тип ***bool*** представляет два логических значения: "***истина***" и "***ложь***". Эти логические

значения обозначаются в ***C#*** зарезервированными словами ***true*** и ***false*** соответственно. Кроме того, в ***C#*** не определено взаимное преобразование логических и целых значений. Например, ***1*** не преобразуется в значение ***true***, а ***0*** — в значение ***false***.

var=========================================================================

Такая переменная называется *неявно типизированной.*

Неявно типизированная переменная объявляется с помощью ключевого слова var

и должна быть непременно инициализирована. Для определения типа этой переменной

компилятору служит тип ее инициализатора, т.е. значения, которым она инициализируется.

Рассмотрим такой пример.

***var е = 2.7183;***

В данном примере переменная е инициализируется литералом с плавающей

точкой, который по умолчанию имеет тип ***double***, и поэтому она относится к типу

***double***. Если бы переменная ***е*** была объявлена следующим образом:

***var е = 2.7183F;***

то она была бы отнесена к типу ***float***.

Тип ***var*** в C# - как ***auto*** в C++.

И последнее замечание: одновременно можно объявить только одну неявно типизированную

переменную. Поэтому объявление

***var s1 = 4.0, s2 = 5.0; // Ошибка!***

является неверным и не может быть скомпилировано. Ведь в нем предпринимается

попытка объявить обе переменные, ***s1*** и ***s2***, одновременно.