Для управления форматированием числовых данных служит другая форма метода

WriteLine(), позволяющая встраивать информацию форматирования, как показано

ниже.

WriteLine(*"форматирующая строка", arg0, arg1, ...* , *argN*);

В этой форме аргументы метода WriteLine() разделяются запятой, а не знаком +.

А *форматирующая строка* состоит из двух элементов: обычных печатаемых символов,

предназначенных для вывода в исходном виде, а также спецификаторов формата. Последние

указываются в следующей общей форме:

***{argnum, width: fmt}***

где ***argnum*** *—* номер выводимого аргумента, начиная с нуля; ***width*** *—* минимальная

ширина поля; ***fmt*** *—* формат. Параметры ***width***и ***fmt***являются необязательными.

Начнем с самого простого примера. При выполнение оператора Console.WriteLine("В феврале {0} или {1} дней.", 28, 29); получается следующий результат.

В феврале 28 или 29 дней

Как видите, значение 28 подставляется вместо спецификатора {0}, а значение 29 —

вместо спецификатора {1}.

Ниже приведен видоизмененный вариант предыдущего оператора, в котором указывается

ширина полей.

***Console.WriteLine("В феврале {0,10} или {1,5} дней.", 28, 29);***

Выполнение этого оператора дает следующий результат.

***В феврале 28 или 29 дней.***

Как видите, неиспользуемые части полей заполнены пробелами. Напомним, что

*минимальная* ширина поля определяется параметром *width.* Если требуется, она может

быть превышена при выводе результата.

using System;

class DisplayOptions

{

static void Main()

{

int i;

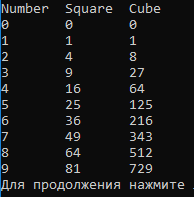
Console.WriteLine("Number\tSquare\tCube");

for (i = 0; i < 10; i++)

Console.WriteLine("{0}\t{1}\t{2}", i, i\*i, i\*i\*i);

}

}



В приведенных выше примерах сами выводимые значения не форматировались.

Но ведь основное назначение спецификаторов формата — управлять внешним видом

выводимых данных. Самый простой способ указать формат данных —

описать шаблон, который будет использоваться в методе ***WriteLine().*** Для этого указывается

образец требуемого формата с помощью символов #, обозначающих разряды

чисел. Кроме того, можно указать десятичную точку и запятые, разделяющие цифры.

Ниже приведен пример более подходящего вывода результата деления 10 на 3.

Console.WriteLine("Деление 10/3 дает: {0:#.##}", 10.0/3.0);

Выполнение этого оператора приводит к следующему результату.

Деление 10/3 дает: 3.33

В данном примере шаблон #.## указывает методу ***WriteLine()*** отобразить два

десятичных разряда в дробной части числа.

Рассмотрим еще один пример. Оператор

***Console.WriteLine("{0:###,###.##}", 123456.56);***

дает следующий результат.

***123,456.56***

Для вывода денежных сумм, например, рекомендуется использовать спецификатор

формата С.

***decimal balance;***

***balance = 12323.09m;***

***Console.WriteLine("Текущий баланс равен {0:С}" , balance);***

Результат выполнения этого фрагмента кода выводится в формате денежных сумм,

указываемых в долларах США.

***Текущий баланс равен $12,323.09***