

## Практическая работа № 5

**Тема:** Ввод-вывод информации. Форматы вывода значений величин.

**Цель:** познакомиться с различными форматами вывода значений; отработать практические навыки вывода значений в разных форматах.

**Задание:**

1. Изучить теоретический материал по данной теме.
2. Составить конспект в рабочей тетради.
3. В среде разработки набрать текст кодов *Примера 1* и *Примера 2*.
4. Откомпилировать коды, проверить правильность их работы.
5. Сдать задание преподавателю, продемонстрировав работу проектов.

### 1. Краткие теоретические сведения

Вывод, производимый методами `System.Console.Write()` и `System.Console.WriteLine()`, можно форматировать. Форматирование позволяет указывать формат целых чисел, чисел с плавающей точкой и других типов данных.

### Управление форматом числовых данных

Пусть в программе определена переменная типа `int` с именем `value`:

```
int value = 250;
```

До этого момента переменные выводились следующим образом:

```
System.Console.WriteLine ("value =" + value);
```

Результат вывода:    `value = 250`

Можно вывести значение `value`, используя требуемое число позиций (например 5):

```
System.Console.WriteLine("value = {0, 5}", value);
```

Первое число в фигурных скобках означает номер переменной – это 0, что соответствует первой переменной `value` в списке метода `System.Console.WriteLine()`. Второе число в фигурных скобках означает количество позиций, отведенное для отображения переменной. В данном примере оно равно 5. При выводе переменной длина ее представления будет дополнена пробелами слева. Если количество позиций меньше чем число знаков переменной, то оно будет выведено без форматирования.

Можно задать форматирование для вывода каждой переменной:

```
int a = -12;  
int b = 20;  
System.Console.WriteLine("a = {0, 4}, b = {1, 3}",  
a, b);
```

Результат вывода: a = -12, b = 20

Форматированный вывод чисел с плавающей точкой немного более сложный. Предположим, определена переменная типа double с именем myDouble: double myDouble = 1234.56789;

Следующий пример выводит значения myDouble, отведя под него десять знакомест, и округлив его до трех цифр после запятой:

```
System.Console.WriteLine("myDouble = {0, 10:f3}",  
myDouble);
```

Символы f3 в этом примере означают, что значение выводится как число с плавающей точкой (символ f), в дробной части будет выведено три цифры.

Точно такое же форматирование можно применять для типов float и decimal. Например:

```
float myFloat = 1234.56789f;  
System.Console.Write("myFloat = {0, 10:f3}",  
myFloat);  
decimal myDecimal = 1234.56789m;  
System.Console.Write("myDecimal = {0, 10:f3}",  
myDecimal);
```

Результат вывода:

```
myFloat = 1234.568;  
myDecimal 1234.568;
```

В списке аргументов методов WriteLine или Write задается строка вида {n, w: спецификатор k} – где n определяет номер идентификатора из списка аргументов метода WriteLine, спецификатор – определяет формат данных, w – целая константа без знака, задает количество символов (длину поля), а k – количество позиций для дробной части значения идентификатора.

Для каждого типа данных существует своя форма представления. Данные сведены в таблицу 2.1.

Таблица 2.1

Тип данных		Форма
Числовые	Целые	W
	Вещественные с фиксированной точностью	W:Fk
	Вещественные в экспоненциальном формате	W:Ek
Логические		W
Символьные		W

Символы форматирования F, E (другие символы форматирования приведены в табл. 2.2) – определяют тип и характеристики объектов ввода-вывода. Параметр w – целая константа без знака, задает количество символов (длину поля), отводимых для ввода-вывода объекта. Параметр k – целая константа без знака определяет для числовых данных:

- количество позиций, для цифр в дробной части числа (форма F);
- количество позиций для цифр, в дробной части мантиссы числа (форма E или G).

В качестве спецификаторов могут использоваться следующие значения:

Таблица 2.2

Символ	Формат	Значение
С или с	Денежный. По умолчанию ставит знак р. Изменить его можно с помощью объекта NumberFormatInfo	Задается количество десятичных разрядов.
Д или d	Целочисленный (используется только с целыми числами)	Задается минимальное количество цифр. При необходимости результат дополняется начальными нулями
Е или е	Экспоненциальное представление чисел	Задается количество символов после запятой. По умолчанию используется 6
Ф или f	Представление чисел с фиксированной точкой	Задается количество символов после запятой
Г или g	Общий формат (или экспоненциальный, или с фиксированной точкой)	Задается количество символов после запятой. По умолчанию выводится целая часть

N или n	Стандартное форматирование с использованием запятых и пробелов в качестве разделителей между разрядами	Задается количество символов после запятой. По умолчанию – 2, если число целое, то ставятся нули
X или x	Шестнадцатеричный формат	
P или p	Процентный	

**Пример 1.** Форматированный вывод данных различного типа.

```
public static void Main()
{
    int a = -14;
    float c = -0.00151f;
    double i = 1234.56789;
    bool l=false;
    string name="Petrov";
    System.Console.WriteLine("name = {0, 6}, l = {1, 4}", name, l);
    System.Console.WriteLine("a ={0, 4}, c = {1,10:f5}, i = {1,20:e8}", a, c, i);
    System.Console.WriteLine(" ");
    System.Console.WriteLine("Для выхода нажмите на Enter");
    System.Console.ReadLine();
}
```

```
file:///c:/C#/Example/ConsoleApplication6/ConsoleApplication6/bin/Debug/ConsoleApplication6.EXE
name = Petrov, l = False
a = -14, c = -0,00151, i = -1,50999997e-003
Для выхода нажмите на Enter
_
```



**Пример 2.** Ввод в диалоге и форматированный вывод данных различного типа.

```
public static void Main()
{
    int a;        // = -14;
    float c;      // = -0.00151f;
    double i;     // = 1234.56789;
    bool l;       // = false;
    string name;  //="Petrov";
    Console.WriteLine("Input фамилию ");
    name = Console.ReadLine();
    Console.WriteLine("Input a");
    a = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Input c");
    c = Convert.ToSingle(Console.ReadLine());
    Console.WriteLine("Input i");
    i = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Input l");
    l = Convert.ToBoolean(Console.ReadLine());

    System.Console.WriteLine(" Результаты
форматирования \n name = {0, 6}, l = {1, 4}",
name, l);
    System.Console.WriteLine("a ={0, 4}, c =
{1,10:f5}, i = {2,20:e8}", a,c,i);
    System.Console.WriteLine(" ");
    System.Console.WriteLine("Для выхода нажмите
на Enter");
    System.Console.ReadLine();
}
```

file:///c:/Example/ConsoleApplication6/ConsoleApplication6/bin/Debug/ConsoleApplication6.EXE

```
Input фамилию
Petrov
Input a
-14
Input c
-0,00151
Input i
1234,56789
Input l
false
Результаты форматирования
name = Petrov, l = False
a = -14, c = -0,00151, i = 1,23456789e+003
Для выхода нажмите на Enter
_
```