

Основы программирования

Синтаксис языка C#

Простейший ввод-вывод

Пример 1. Простейший вывод

```
int a = 10;  
double x = 3.14, y = 6.28;  
string s = "Мама мыла раму";
```

```
Console.WriteLine(s); // вывод 1  
Console.WriteLine("a = " + a); // вывод 2  
Console.WriteLine("x=" + x + "; y=" + y); // вывод 3
```

```
Мама мыла раму  
a = 10  
x = 3,14; y = 6,28
```

Нумерация параметров

```
Console.WriteLine("x={0}; y={4}", x, y);  
Console.WriteLine("y={1}", x, y, z);
```

Перегруженные методы

Методы с одинаковыми именами и различными параметрами называется перегруженными.

```
Console.WriteLine("x=" + x + "; y=" + y);  
Console.WriteLine("x={0}; y={1}", x, y);
```

| |
|---------------------------|
| <code>x = 2; y = 4</code> |
|---------------------------|

Три варианта вывода. Интерполяция

(последние версии языка)

```
Console.WriteLine("x=" + x + "; y="+ y);
```

```
Console.WriteLine("x={0}; y={1}", x, y);
```

```
Console.WriteLine($"x={x}; y={y}"); //интерполяция
```

Форматирование. Параметры формата

{номер элемента, ширина поля: формат}

Например:

{0, 5:0.000}

{0, 5:0.###}

{0, 5:0.##0}

{0, 5:#.##0},

где

0 — обязательный элемент,

— необязательный элемент.

Форматирование. Параметры формата

```
double x = 13.1;  
Console.WriteLine("(1) x = {0}", x);  
Console.WriteLine("(2) x = {0,5:0.000}", x);  
Console.WriteLine("(3) x = {0,5:0.###}", x);  
Console.WriteLine("(4) x = {0,5:0.##0}", x);  
Console.WriteLine("(5) x = {0,5:#.##0}", x);  
Console.ReadLine();
```

```
(1) x = 3,1  
(2) x = 3,100  
(3) x = 3,1  
(4) x = 3,100  
(5) x = 3,100
```

Стандартные форматы

Методы записи класса Console

`Console.WriteLine()` осуществляет переход на следующую строку

`Console.Write()` не осуществляет переход на следующую строку.

Чтение с консоли

- символы считываются в строковую переменную;
- полученная строковая переменная преобразуется к значению искомого типа.

Пример чтения с консоли

```
Console.Write("Введите целое число");  
string s = Console.ReadLine();  
int a = Convert.ToInt32(s);
```

```
Console.Write("Введите вещественное число");  
string s = Console.ReadLine();  
double x = Convert.ToDouble(s);
```

или так

```
double x = Convert.ToDouble(Console.ReadLine());
```

Методы чтения класса Console

`Console.ReadLine()`

`Console.Read()`

Метод ReadLine

- Считывает все символы из буфера клавиатуры до символа `Enter` в строковую переменную.
- При этом символ перевода строки удаляется из буфера, а сам буфер клавиатуры очищается.

Метод Read

- Читает по одному символу из входной строки и возвращает в качестве результата код прочитанного символа.
- Если символ не прочитан, то возвращает “-1”.
- Требуется явное преобразование к типу `char`.
- Посимвольный ввод применяется редко.
- Не очищает буфер.

Пример

```
char c;  
Console.WriteLine("Введите символ");  
c = Convert.ToChar(Console.Read());  
Console.ReadLine();  
Console.WriteLine("Вы ввели символ:" + c);  
Console.ReadLine();
```

Разделитель целой и дробной части

В одних культурах разделителем между целой и дробной частью является точка (США, Великобритания...), в других - запятая (Россия, Германия...).

При вводе важно учитывать какой разделитель выбран в настройках операционной системы.

Например, если в «региональных настройках» разделителем выбрана запятая, значит, надо вводить дробные числа с разделителем запятой.

Если точка, то разделителем целой и дробной части при вводе будет точка.