Консольный ввод-вывод

Консольный вывод

Для вывода информации на консоль мы уже использовали встроенный метод **Console.WriteLine**. То есть, если мы хотим вывести некоторую информацию на консоль, то нам надо передать ее в метод Console.WriteLine:

```
string hello = "Привет мир";
Console.WriteLine(hello);
Console.WriteLine("Добро пожаловать в C#!");
Console.WriteLine("Пока мир...");
Console.WriteLine(24.5);
```

Консольный вывод:

```
Привет мир!
Добро пожаловать в C#!
Пока мир...
24,5
```

Нередко возникает необходимость вывести на консоль в одной строке значения сразу нескольких переменных. В этом случае мы можем использовать прием, который называется интерполяцией:

```
string name = "Tom";
int age = 34;
double height = 1.7;
Console.WriteLine($"Имя: {name}  Возраст: {age}  Рост: {height}м");
```

Для встраивания отдельных значений в выводимую на консоль строку используются фигурные скобки, в которые заключается встраиваемое значение. Это можем значение переменной (`{name}`) или более сложное выражение (например, операция сложения `{4 + 7}`). А перед всей строкой ставится знак доллара $.

При выводе на консоль вместо помещенных в фигурные скобки выражений будут выводиться их значения:

```
Имя: Tom  Возраст: 34  Рост: 1,7м
```

Есть другой способ вывода на консоль сразу нескольких значений:

```
string name = "Tom";
int age = 34;
double height = 1.7;
Console.WriteLine("Имя: {0}  Возраст: {2}  Рост: {1}м", name, height, age);
```

Здесь мы видим, что строка в `Console.WriteLine` содержит некие числа в фигурных скобках: {0}, {1}, {2}. Это плейсхолдеры, вместо которых при выводе строки на консоль будут подставляться некоторые значения. Подставляемые значения указываются после строки через запятую.

При этом важен порядок подобных плейсхолдеров. Например, в данном случае после строки первой указана переменная name, потом height и потом age. Поэтому значение переменной name будет вставляться вместо первого плейсхолдера - {0} (нумерация начинается с нуля), height - вместо {1}, а age - вместо {2}. Поэтому в итоге пи выводе на консоль строка

```
"Имя: {0}  Возраст: {2}  Рост: {1}м"
```

будет заменена на

```
"Имя: Tom  Возраст: 34  Рост: 1,7м"
```

Console.Write

Кроме `Console.WriteLine()` можно также использовать метод **Console.Write()**, он работает точно так же за тем исключением, что не добавляет переход на следующую строку, то есть последующий консольный вывод будет выводиться на той же строке.

```
string name = "Tom";
int age = 34;
double height = 1.7;
Console.Write($"Имя: {name}  Возраст: {age}  Рост: {height}м");
```

Консольный ввод

Кроме вывода информации на консоль мы можем получать информацию с консоли. Для этого предназначен метод **Console.ReadLine()**. Он позволяет получить введенную строку.

```
Console.Write("Введите свое имя: ");
string? name = Console.ReadLine();
Console.WriteLine($"Привет {name}");
```

В данном случае все, что вводит пользователь, с помощью метода Console.ReadLine() передается в переменную name.

Пример работы программы:

```
Введите свое имя: Том
Привет Том
```

Особенностью метода `Console.ReadLine()` является то, что он может считать информацию с консоли только в виде строки. Кроме того, возможная ситуация, когда для метода Console.ReadLine не окажется доступных для считывания строк, то есть когда ему нечего считывать, он возвращаает значение **null**, то есть, грубо говоря, фактически отсутствие значения. И чтобы отразить эту ситуацию мы определяем переменную `name`, в которую получаем ввод с консоли, как переменную типа **string?**. Здесь `string` указывает, что переменная может хранить значения типа string, то есть строки. А знак вопроса **?** указывает, что переменная также может хранить значение **null**, то есть по сути не иметь никакого значения. Далее мы более подробно разберем `null` и как с ним работать.

Однако, может возникнуть вопрос, как нам быть, если, допустим, мы хотим ввести возраст в переменную типа int или другую информацию в переменные типа double или decimal? По умолчанию платформа .NET предоставляет ряд методов, которые позволяют преобразовать различные значения к типам int, double и т.д. Некоторые из этих методов:

- **Convert.ToInt32()** (преобразует к типу int)
- **Convert.ToDouble()** (преобразует к типу double)
- **Convert.ToDecimal()** (преобразует к типу decimal)

Пример ввода значений:

```
Console.Write("Введите имя: ");
string? name = Console.ReadLine();

Console.Write("Введите возраст: ");
int age = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите рост: ");
double height = Convert.ToDouble(Console.ReadLine());
```

```
9
10  Console.Write("Введите размер зарплаты: ");
11  decimal salary = Convert.ToDecimal(Console.ReadLine());
12
13  Console.WriteLine($"Имя: {name}  Возраст: {age}  Рост: {height}м  Зарплата: {sal
```

При вводе важно учитывать текущую операционную систему. В одних культурах разделителем между целой и дробной частью является точка (США, Великобритания...), в других - запятая (Россия, Германия...). Например, если текущая ОС - русскоязычная, значит, надо вводить дробные числа с разделителем запятой. Если локализация англоязычная, значит, разделителем целой и дробной части при вводе будет точка.

Пример работы программы:

```
Введите имя: Том
Введите возраст: 25
Введите рост: 1,75
Введите размер зарплаты: 300,67
Имя: Том  Возраст: 25  Рост: 1,75м  Зарплата: 300,67$
```