

# Конструкция `if..else` и тернарная операция

Условные конструкции - один из базовых компонентов многих языков программирования, которые направляют работу программы по одному из путей в зависимости от определенных условий. Одной из таких конструкций в языке программирования C# является конструкция `if..else`

Конструкция `if/else` проверяет истинность некоторого условия и в зависимости от результатов проверки выполняет определенный код.

Ее простейшая форма состоит из блока `if`:

```
1  if(условие)
2  {
3      выполняемые инструкции
4  }
```

После ключевого слова `if` ставится условие. Условие должно представлять значение типа **bool**. Это может быть непосредственно значение типа `bool` или результат условного выражения или другого выражения, которое возвращает значение типа `bool`. И если это условие истинно (равно `true`), то срабатывает код, который помещен далее после условия внутри фигурных скобок.

Например:

```
1  int num1 = 8;
2  int num2 = 6;
3  if(num1 > num2)
4  {
5      Console.WriteLine($"Число {num1} больше числа {num2}");
6  }
```

В данном случае у нас первое число больше второго, поэтому выражение `num1 > num2` истинно и возвращает `true`, следовательно, управление переходит к строке `Console.WriteLine("Число {num1} больше числа {num2}");`

Если блок `if` содержит одну инструкцию, то мы можем его сократить, убрав фигурные скобки:

```
1  int num1 = 8;
2  int num2 = 6;
3  if (num1 > num2)
4      Console.WriteLine($"Число {num1} больше числа {num2}");
5
```

```
6 // или так
7 if (num1 > num2) Console.WriteLine($"Число {num1} больше числа {num2}");
```

Также мы можем соединить сразу несколько условий, используя логические операторы:

```
1 int num1 = 8;
2 int num2 = 6;
3 if (num1 > num2 && num1==8)
4 {
5     Console.WriteLine($"Число {num1} больше числа {num2}");
6 }
```

В данном случае блок `if` будет выполняться, если `num1 > num2` равно `true` и `num1==8` равно `true`.

### *Выражение else*

Но что, если мы захотим, чтобы при несоблюдении условия также выполнялись какие-либо действия? В этом случае мы можем добавить блок `else`:

```
1 int num1 = 8;
2 int num2 = 6;
3 if (num1 > num2)
4 {
5     Console.WriteLine($"Число {num1} больше числа {num2}");
6 }
7 else
8 {
9     Console.WriteLine($"Число {num1} меньше числа {num2}");
10 }
```

Блок `else` выполняется, если условие после `if` ложно, то есть равно `false`. Если блок `else` содержит только одну инструкцию, то оять же мы можем его сократить, убрав фигурные скобки:

```
1 int num1 = 8;
2 int num2 = 6;
3 if (num1 > num2)
4     Console.WriteLine($"Число {num1} больше числа {num2}");
```

```
5     else

6         Console.WriteLine($"Число {num1} меньше числа {num2}");

else if
```

Но в примере выше при сравнении чисел мы можем насчитать три состояния: первое число больше второго, первое число меньше второго и числа равны. Используя конструкцию `else if`, мы можем обрабатывать дополнительные условия:

```
1     int num1 = 8;

2     int num2 = 6;

3     if (num1 > num2)

4     {

5         Console.WriteLine($"Число {num1} больше числа {num2}");

6     }

7     else if (num1 < num2)

8     {

9         Console.WriteLine($"Число {num1} меньше числа {num2}");

10    }

11    else

12    {

13        Console.WriteLine("Число num1 равно числу num2");

14    }
```

При необходимости можно добавить несколько выражений `else if`:

```
1     string name = "Alex";

2

3     if (name == "Tom")

4         Console.WriteLine("Вас зовут Tomas");

5     else if (name == "Bob")

6         Console.WriteLine("Вас зовут Robert");

7     else if (name == "Mike")

8         Console.WriteLine("Вас зовут Michael");

9     else

10        Console.WriteLine("Неизвестное имя");
```

## Тернарная операция

Тернарную операция также позволяет проверить некоторое условие и в зависимости от его истинности выполнить некоторые действия. Она имеет следующий синтаксис:

```
1  [первый операнд - условие] ? [второй операнд] : [третий операнд]
```

Здесь сразу три операнда. В зависимости от условия тернарная операция возвращает второй или третий операнд: если условие равно `true`, то возвращается второй операнд; если условие равно `false`, то третий. Например:

```
1  int x=3;
2  int y=2;
3
4  int z = x < y ? (x+y) : (x-y);
5  Console.WriteLine(z);    // 1
```

Здесь первый операнд (то есть условие) представляет выражение `x < y`. Если оно равно `true`, то возвращается второй операнд - `(x+y)`, то есть результат операции сложения. Если условие равно `false`, то возвращается третий операнд - `(x-y)`.

Результат тернарной операции (то есть второй или третий операнд в зависимости от условия) присваивается переменной `z`.