

Основы программирования

Синтаксис языка C#

Операторы отношений и
логические операторы

Условные выражения

Такие операции возвращают логическое значение, то есть значение типа `bool`:

- `true`, если выражение истинно, и
- `false`, если выражение ложно.

К подобным операциям относятся операции сравнения и логические операции.

Операторы сравнения

В операциях сравнения сравниваются два операнда и возвращается значение типа `bool`: `true`, если выражение верно, и `false`, если выражение неверно.

Поддерживаются всеми целочисленными типами и типами с плавающей запятой.

Операторы сравнения

`==` (сравнивает два операнда на равенство. Если они равны, то операция возвращает `true`, если не равны, то возвращается `false`)

```
int a = 10;
```

```
int b = 4;
```

```
bool c = a == b;           // false
```

Операторы сравнения

!= (сравнивает два операнда и если они равны, то операция возвращает `false`, если не равны, то возвращается `true`)

```
int a = 10;
```

```
int b = 4;
```

```
bool c = a != b;           // true
```

```
bool d = a!=10;           // false
```

Операторы сравнения

`<` (операция "меньше чем". Возвращает `true`, если первый операнд меньше второго, и `false`, если первый операнд больше второго)

```
int a = 10;
```

```
int b = 4;
```

```
bool c = a < b;           // false
```

Операторы сравнения

`<=` (операция “меньше или равно”. Сравнивает два операнда и возвращает `true`, если первый операнд меньше или равен второму, иначе возвращается `false`)

```
int a = 10;  
int b = 4;  
bool c = a <= b;      // false  
bool d = a <= 25;     // true
```

Операторы сравнения

`>=` (операция “больше или равно”. Сравнивает два операнда и возвращает `true`, если первый операнд больше или равен второму, иначе возвращается `false`)

```
int a = 10;  
int b = 4;  
bool c = a >= b;      // true  
bool d = a >= 25;     // false
```


Приоритет операций сравнения

Операции

- $<$
- $>$
- $<=$
- $>=$

имеют больший приоритет, чем

- $==$
- $!=$

Логические операторы

В C# определены логические операторы, которые также возвращают значение типа `bool`.

В качестве операндов они принимают значения типа `bool`.

Как правило, применяются к отношениям и объединяют несколько операций сравнения.

Логические операторы

| (операция логического сложения или логическое ИЛИ. Возвращает `true`, если хотя бы один из операндов возвращает `true`)

```
bool x1 = (5 > 6) | (4 < 6);  
          // 5 > 6 - false, 4 < 6 - true,  
          поэтому возвращается true
```

```
bool x2 = (5 > 6) | (4 > 6);  
          // 5 > 6 - false, 4 > 6 - false,  
          поэтому возвращается false
```

Логические операторы

& (операция логического умножения или логическое И. Возвращает `true`, если оба операнда одновременно равны `true`.)

```
bool x1 = (5 > 6) & (4 < 6);  
           // 5 > 6 - false, 4 < 6 - true,  
           поэтому возвращается false
```

```
bool x2 = (5 < 6) & (4 < 6);  
           // 5 < 6 - true, 4 < 6 - true,  
           поэтому возвращается true
```

Логические операторы

|| (Операция логического сложения. Возвращает `true`, если хотя бы один из операндов возвращает `true`)

```
bool x1 = (5 > 6) | (4 < 6);  
           // 5 > 6 - false, 4 < 6 - true,  
           поэтому возвращается true
```

```
bool x2 = (5 > 6) | (4 > 6);  
           // 5 > 6 - false, 4 > 6 - false,  
           поэтому возвращается false
```

Логические операторы

&& (операция логического умножения. Возвращает `true`, если оба операнда одновременно равны `true`)

```
bool x1 = (5 > 6) & (4 < 6);  
           // 5 > 6 - false, 4 < 6 - true,  
           поэтому возвращается false
```

```
bool x2 = (5 < 6) & (4 < 6);  
           // 5 < 6 - true, 4 < 6 - true,  
           поэтому возвращается true
```

Логические операторы

- ^ (операция исключающего ИЛИ. Возвращает `true`, если либо первый, либо второй операнд (но не одновременно) равны `true`, иначе возвращает `false`)

```
bool x5 = (5 > 6) ^ (4 < 6);  
          // 5 > 6 - false, 4 < 6 - true,  
          поэтому возвращается true
```

```
bool x6 = (50 > 6) ^ (4 / 2 < 3);  
          // 50 > 6 - true, 4/2 < 3 - true,  
          поэтому возвращается false
```

Логические операторы

! (операция логического отрицания. Производится над одним операндом и возвращает `true`, если операнд равен `false`. Если операнд равен `true`, то операция возвращает `false`:)

```
bool a = true;  
bool b = !a;    // false
```


Выполнение логических операторов

унарный:

! (логическое отрицание)

бинарные:

& (логическое И),

| (логическое ИЛИ)

^ (логическое исключающее ИЛИ)

Эти операторы всегда обрабатывают оба операнда.

Выполнение логических операторов

бинарные:

&&	(условное логическое И)
	(условное логическое ИЛИ).

Эти операторы вычисляют правый операнд, только если это необходимо.

Сокращенные логические операторы

Различие между

- обычной (& и |)
- сокращенной (&& и ||)

версиями логических операторов состоит в том, что при использовании обычной операции всегда вычисляются оба операнда, в случае использования сокращенной версии второй операнд вычисляется только при необходимости.

Сокращенный оператор И также называется *условным И*, а сокращенный ИЛИ — *условным ИЛИ*.

Пример

```
z = x || y;
```

сначала будет вычисляться значение `x`, и если оно равно `true`, то вычисление значения `y` уже смысла не имеет, так как у нас в любом случае уже `z` будет равно `true`. Значение `y` будет вычисляться только в том случае, если `x` равно `false`

```
z = x && y;
```

сначала будет вычисляться значение `x`, и если оно равно `false`, то вычисление значения `y` уже смысла не имеет, так как у нас в любом случае уже `z` будет равно `false`. Значение `y` будет вычисляться только в том случае, если `x` равно `true`

Сокращенные операции

Сокращенные операции `||` и `&&` более удобны в вычислениях, так как позволяют сократить время на вычисление значения выражения, и тем самым повышают производительность.

Операции `|` и `&` больше подходят для выполнения поразрядных операций над числами.

Приоритет логических операций

в порядке убывания приоритета:

- Оператор логического отрицания !
- Оператор логического И &
- Оператор логического исключающего ИЛИ ^
- Оператор логического ИЛИ |
- Условный оператор логического И &&
- Условный оператор логического ИЛИ ||