

Выполнение логических операций

Логические операторы бывают 2 типов:

- Простые (операторы **И** и **ИЛИ**; могут возвращать два значения: **true** (истина) и **false** (ложь));
- Побитовые (позволяют работать с любым **битом** в машинном представлении числа).

Логические
операторы

Оператор	Значение
==	Равенство; истинно, когда значение левого аргумента совпадает со значением правого
!=	Неравенство; противоположное равенству
> <	Больше, меньше; истинно, когда значение левого выражения больше (меньше) значения правого
>= <=	Больше или равно, меньше или равно; истинно, если истиной является > или == (< или ==)
&&	И; истинно, если аргументы и слева, и справа являются истиной
	ИЛИ; истинно, если или левый, или правый аргумент являются истиной
!	НЕ; истинно, если его аргумент принимает

Операторы `==`, `!=`, `>`, `<`, `>=`, `<=`, `&&`, `||` являются операторами сравнения. Оператор равенства используется для проверки двух значений. Например, следующее выражение истинно, если значением `x` является `5`, и ложно во всех других случаях:

```
x == 5;
```

```
x == 10;
```

Оператор
равенства

```
x = 10;
```

Оператор
присваивания

Распространенная ошибка, которую компилятор не посчитает за ошибку и не укажет на нее.

Операторы **&&** (**И**) и **||** (**ИЛИ**) обычно сочетаются с другими логическими операторами.

(x < y) && (y < z)

// истинно, если y больше x и

меньше z

(x < y) || (y < z)

// истинно, если y больше x или

меньше z

Операторы сравнения достаточно взаимозаменяемы.
Например:

$x \leq y$

эквивалентно

$(x < y) \ || \ (x == y)$

$x == y$

эквивалентно

$!(x > y) \ \&\& \ !(x < y)$

В выражении $(x > y) \ \&\& \ (a < b)$ C++ вычисляет первым условие $(x > y)$. В случае если оно **ложно** (для оператора **&&**) или **истинно** (для оператора **||**), выражение $(a < b)$ не вычисляется и не анализируется.

Хранение логических значений

Результат логической операции может быть присвоен переменной типа bool:

```
int n1 = 1;
```

```
int n2 = 2;
```

```
bool b;
```

```
b = (n1 == n2);
```

“Сравни содержимое переменных n1 и n2 и сохрани результат сравнения в переменной b”.

По правилу приоритетов операций выражение

`b = (n1 == n2);` эквивалентно выражению `b = n1 == n2;`

BoolTest

Инструкция `cout.setf (cout.boolalfa);` обеспечивает вывод логического значения в виде строки “true” или “false”, а не в виде 1 или 0, как принято по умолчанию.

Программа запрашивает у пользователя два числовых значения и возвращает результат их сравнения. Пример работы программы:

```
Введите значение 1: 5  
Введите значение 2: 5  
Значение 5 == 5 равно true  
Press any key to continue...
```

Логические операции с вещественными переменными

```
float a = 10.0;
```

```
float b = a / 3;
```

```
a == (b * 3.0); // Равны ли эти значения?
```

Математически $b = 3\frac{1}{3}$

но `float b = 3.333333`

значит `a == (b * 3.0) – false`

Логические операции с вещественными переменными

```
float a = 10.0;
```

```
float b = 100 % 30;
```

```
a == b; // Истинно ли это выражение?
```

Математически $b = 10$

но $100.0 : 30.0 = 3,333333$ (или 3.33347)

```
float b = 100.0 % 30.0 = ?
```

Более безопасное сравнение:

```
float a = 10;
```

```
float b = a / 3;
```

```
float c = b * 3.0;
```

```
(a - c) < 0.0001 && (c - a) < 0.0001;
```

Истинно в случае, если разница между a и b меньше $0,00001$.

Сокращенные вычисления в

C++

`условие1 && условие2`

Если `условие1` ложно, то результат будет ложным независимо от истинности выражения `условие2`.

`условие1 || условие2`

Если `условие1` истинно, то результат будет истинным независимо от истинности выражения `условие2`.

Для экономии времени C++ вычисляет первым `условие1`, и, в случае, если оно ложно (для оператора `&&`) или истинно (для оператора `||`), выражение `условие2` не вычисляется и не анализируется.