

Операции отношения и логические операции

Как я уже писал ранее, условный оператор выполняет те или иные действия в зависимости от выполнения условия, заданного программистом. Давайте обсудим, какие условия можно задать в языке Си.

Сразу же напомним, что компьютер не очень-то сообразительный. В данном случае это проявляется в том, что он понимает только простые условия на да/нет. Но обо всём по порядку.

Самое простое, что умеет делать компьютер, – это сравнивать между собой числа и значения числовых выражений. Для этого предназначены **операции отношения**.

Листинг 1.

```
> - больше
>= - больше либо равно
< - меньше
<= - меньше либо равно
== - равно
!= - не равно
```

Вот посмотрите на некоторые примеры:

Листинг 2.

```
#include <stdio.h>

int main(void) {

    int a = 10, b = 5, c = 2;
    char ch = 'a';

    printf("%d\n", b > c );
    printf("%d\n", a == b*c );
    printf("%d\n", a != b*c );
    printf("%d\n", a < c );
    printf("%d\n", b >= c + 3 );
    printf("%d\n", -1 <= 2 );
    printf("%d\n", ch == 'a' );
    printf("%d\n", ch != 'a' );

    return 0;
}
```

Вы ещё помните, что компьютер понимает только нули и единицы? Так вот, как вы уже догадались:

Правило!

ноль это ложь, а единица – истина.

Вообще, любое число, кроме нуля, интерпретируется как истина. Думаю, дополнительных пояснений эти операции не требуют. Хотя на операцию сравнения обратите особое внимание. Т.к. один знак равно `=` используется для оператора присваивания, то для операции сравнения используется два знака равенства `==`. Будьте внимательны и не забывайте об этом. Это частая ошибка, которую совершают начинающие программисты.

Кроме операций отношения есть ещё и **логические операции**. Они используются для того, чтобы из простых условий создавать сложные. Уже немного испугались? Да не, на самом деле ничего сложного. Сейчас разберёмся.

Логических операций всего три штуки:

Листинг 3.

```
! - логическое НЕ.
&& - логическое И.
|| - логическое ИЛИ.
```

Давайте объясню, как они работают.

Оператор **логическое НЕ** `!`(условие) отрицает выражение, к которому он был применён. Если условие было истиной, оно становится ложью, а если было ложью, то становится истиной. Вот пример:

Листинг 4.

```
#include <stdio.h>

int main(void) {

    int a = 10, b = 5, c = 2;

    printf("%d\n", a == b*c );
    printf("%d\n", !(a == b*c) );
    printf("%d\n", a != b*c );
    printf("%d\n", !(a != b*c) );

    return 0;
}
```

(условие1) `&&` (условие2) - оператор **логическое И**. Возвращает истину, если оба условия истинны. Иначе возвращает ложь.

Примеры:

Листинг 5.

```
#include <stdio.h>
```

```
int main(void) {

    printf("%d\n", 1 && 1 );
    printf("%d\n", 1 && 0 );
    printf("%d\n", 0 && 1 );
    printf("%d\n", 0 && 0 );

    return 0;
}
```

(условие1) || (условие2) - оператор **логическое ИЛИ**. Возвращает истину, если хотя бы одно из условий истина или оба условия сразу истинны. Иначе возвращает ложь.

Примеры:

Листинг 6.

```
#include <stdio.h>

int main(void) {

    printf("%d\n", 1 || 1 );
    printf("%d\n", 1 || 0 );
    printf("%d\n", 0 || 1 );
    printf("%d\n", 0 || 0 );

    return 0;
}
```

Приоритет логических операций

Для логических операций тоже есть определённая очередность выполнения.

- Операции в скобках
- Логическая операция НЕ
- Логическая операция И
- Логическая операция ИЛИ

Включим логические операции в общий список приоритетов операций, который мы составляли для арифметических операций и математических функций.

- Операции в скобках
- Вычисляются функции (например, `sqrt()`, `cos()` и др.)
- Логическая операция НЕ

- умножение, деление, остаток от деления (слева направо)
- сложение, вычитание
- Логическая операция И
- Логическая операция ИЛИ
- Выполняется присваивание

Практика

1. Решите предложенные [задачи](#) с автоматической проверкой решения.

Интернет версия: http://youngcoder.ru/lessons/6/otnosheniya_i_usloviya.php