Глава 3 «Ветвление»

§3.2 Логические операции

Логический тип данных в C++

В программировании нам необходимо знать, какие значения могут принимать логические выражения. Для этого в языке C++ существует специальный тип данных – **bool**. Допустимыми значениями этого типа являются только **true либо 1** (логическое выражение истинно) и **false либо 0** (логическое выражение ложно), при этом других значений у переменной данного типа бы кет. П типа boo т в памя **1 ба**

```
if (7 > 4){

}
```

Значение логического выражения ложно (false)

```
if (5 != 8){

}
```

Значение логического выражения истинно (true)

```
int a;
if (a < 3){

}
```

Значение логического выражения зависит от переменной a, если a меньше 3, то true, если больше или равно 3, то false

```
bool b;
if (b == true){

}
```

Значение логического выражения истинно (true), когда значение переменной b

Логические операторы в C++

Логические операторы не выполняют обычных арифметических преобразований. Вместо этого они оценивают каждое логическое выражение на истинность. Результатом логической операции является значения **true** или **false**. Операторы бывают **унарные** (1 логическое выражение) и **бинарные** (2 логических выражения).

Виды логических операторов:

1) **!** – унарный оператор **НЕ**
2) **&&** - бинарный оператор **И**
3) **||** - бинарный оператор **ИЛИ**
4) **^** - бинарный оператор **исключающее ИЛИ**

Логический оператор НЕ

Унарный логический оператор НЕ (обозначается **!**) изменяет значение логического выражения на противоположное. Он возвращает значение true, **когда значение логического выражен** false **оборот.**

```
int a = 3;
int b = 4;
if (!(b < a)){

}
```

Значение логического выражения в скобках ложно, значит само выражение = true

```
if (!(7 > 4)){

}
```

Значение логического выражения в скобках истинно, значит само выражение = false

Логический оператор И

Бинарный логический оператор И (обозначается **&&**) возвращает значение true, если **оба логических выражения истинны**. Если любое из выражений ложно, возвращаемое значение будет false. Если первое логическое выражение имеет значение false, то второе н

```
int a = 3;
int b = 4;
if (7 > 4 && a < b){

}
```

Значение логического выражения = true, так как оба выражения истинны

```
int a = 3;
int b = 4;
if (4 > 7 && a < b){

}
```

Значение логического выражения = false, так как первое выражение ложно, а второе истинно.

Логический оператор ИЛИ

Бинарный логический оператор ИЛИ (обозначается **||**) возвращает значение true, если **хотя бы одно из логических выражения истинно**. Значение false присваивается только тогда, когда оба выражения ложны. Если первое логическое выражение имеет значение true, второе не вычисляется.

```
int a = 3;
int b = 4;
if (4 > 7 || a < b){

}
```

Значение логического выражения = true, так как первое выражение ложно, а второе истинно.

```
int a = 3;
int b = 4;
if (4 > 7 || b < a){

}
```

Значение логического выражения = false, так как оба выражения ложные.

Логический оператор исключающее ИЛИ

Бинарный логический оператор исключающее ИЛИ (обозначается ^) возвращает значение true, если **только одно из логических выражения истинно**. Значение false присваивается, когда либо оба выра[illegible]ы, либо оба вы[illegible]ы.

```
int a = 3;
int b = 4;
if (7 > 4 ^ a < b){

}
```

Значение логического выражения = false, так как оба выражения истинны

```
int a = 3;
int b = 4;
if (7 > 4 ^ b < a){

}
```

Значение логического выражения = true, так как первое выражение истинно, а второе ложно.

Приоритет логических операторов

Порядок выполнения логических операций в выражениях зависит от порядка, их приоритета и скобок. Высший приоритет имеют операторы, **заключенные в скобки**, затем оператор **! (НЕ)**, далее **&& (И)** и самый низкий приоритет у оператора **|| (ИЛИ).**

```
int a = 3;
int b = 4;
if (a < b || 7 < 3 && (!a || !b)){

}
```

Сначала выполняются операторы ! в скобках, затем оператор || в скобках, затем оператор && вне скобок и в последнюю очередь оператор || вне скобок