

Логические операции	
&&	и
	или
!	отрицание

Операции сравнения	
==	равенство
!=	неравенство
>	больше
>=	больше или равно
<	меньше
<=	меньше или равно

Замечание 1. В языке C++, как и в почти любом языке программирования, нельзя соединять неравенства в цепочки так, как принято в математике. Например, если мы хотим проверить неравенство $1 < x < 5$, нельзя просто написать в программе $1 < x < 5$. Последнее выражение будет понято компилятором как $(1 < x) < 5$, где результатом выражения в скобках будет 0, если это неравенство ложно, и 1, если оно истинно. В итоге, например, при $x=7$ результатом выражения в скобках будет 1, поскольку это неравенство истинно, и результатом всего выражения тоже будет 1, поскольку $1 < 5$, тогда как в математике такое неравенство ложно (так как $7 > 5$).

Если же нам надо проверить два неравенства одновременно, нужно воспользоваться логической операцией и, и правильное условие в программе будет выглядеть так: $1 < x \ \&\& \ x < 5$.

Замечание 2. В арифметических выражениях операнды вычисляются в том порядке, который удобнее компилятору, так что предсказать его заранее трудно. Логические выражения отличаются от арифметических тем, что в них порядок вычисления операндов строго определен --- слева направо, причем если значения первого операнда достаточно для определения результата всей логической операции, то второй не вычисляется. Такое соглашение о вычислении логических выражений называется *сокращенным вычислением*.

Например, если у нас есть логическое выражение $1 < x \ \&\& \ x < 5$ и переменная x имеет значение 0, то первое неравенство уже неверно. При этом, каково бы ни было значение второго неравенства, результат всего выражения будет ложь, и поэтому в данном случае второе неравенство проверяться не будет.

Это бывает удобно в тех случаях, когда второе выражение определено не всегда. Например, если у нас есть условие $x \neq 3 \ \&\& \ 1/(x-3) < 20$, то при $x=3$ оно тоже вполне может быть вычислено, потому что в этом случае результат первой части будет ложь и вторая (не имеющая в данном случае смысла) не будет проверяться.