Условные операторы

Понятие логического выражения

Логическое выражение (тип *boolean*) принимает всего два значения: false (ложь) и true (истина).

Пример:

(x>10) && (y<x+2)

Группы операций для логических выражений (в порядке уменьшения приоритета):

1) арифметические операции и функции
2) операции сравнения:

>, >=, <, <=, == , !=

3) логические операции:

- двухместные: &&, ||, ^
- одноместные: !

Логическое выражение вычисляется слева направо с учетом приоритетов.

&& (AND) – логическое «и» (логическое умножение). Принимает значение true, если оба операнда имеют это значение.

Пример:

(2>3) && (3<5) => false

false *true*

|| (OR) – логическое «или» (логическое сложение). Принимает значение true, если значение хотя бы одного операнда истина.

Пример:

(2>3) || (3<5) => true

^ (XOR) – исключающее «или» (сложение по модулю 2). Принимает значение true, если значения ее операндов различны (только один операнд имеет значение true)

Таблица истинности AND, OR, XOR

O1, O2 – операнды

0 – false

1 – true

O1	**O2**	**&&**	**||**	**^**
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

! (NOT) – логическое «не» (операция отрицания). Принимает значение true, если значение ее операнда ложно.

Пример:

!(3>5) => true

Таблица истинности NOT

Операнд	**!**
0	1
1	0

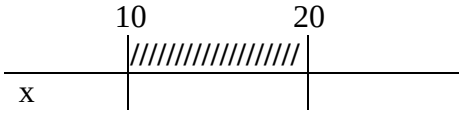
Приоритеты операций

В порядке уменьшения приоритета

	Высший	***Комментарий***
1	( ) [ ]	*Группирующие скобки, обращение к элементу по индексу*
2	! ++ --	
3	* / %	
4	+ -	
5	>> <<	
6	>= > <= <	
7	== !=	
8	&	*Побитовое И*
9	^	
10	|	*Побитовое ИЛИ*
11	&&	
12	||	
13	? :	*Тернарный оператор*
14	= += -= *= /= %=	
	Низший	

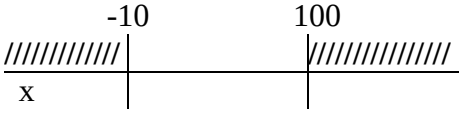
Примеры:

1) Принадлежность интервалу



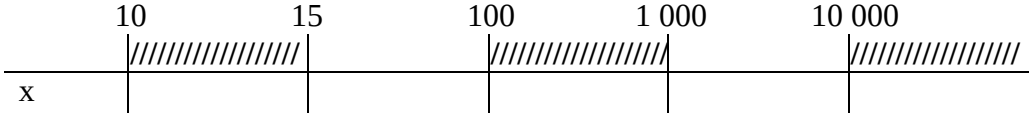
(x>=10) && (x<=20)

2) Принадлежность интервалу



(x<=-10) || (x>=100)

3) Принадлежность интервалу



((x>=10) && (x<=15)) || ((x>100) && (x<=1000)) || (x>=10 000)

4) Побитовое операции

156 << 4 & 120 | 11 >> 3

1. $156_{10} = 128 + 16 + 8 + 4 = 2^7 + 2^4 + 2^3 + 2^2 = 1001\ 1100_2$
   $1001\ 1100_2$ << 4 = 1001 1100 0000

2. $11_{10} = 8 + 2 + 1 = 2^3 + 2^1 + 2^0 = 1011_2$
$1011_2 >> 3 = 1$

3. $120_{10} = 64 + 32 + 16 + 8 = 2^6 + 2^5 + 2^4 + 2^3 = 111\ 1000_2$

&
1001 1100 0000
0000 0111 1000
0000 0100 0000

4. |
100 0000
000 0001
100 0001

$100\ 0001_2 = 2^6 + 2^1 = 64 + 1 = 65$

Примечание:
&& и || вычисляют выражение до получения результата: вычисления прекращаются, когда результат становится очевидным. Например:
a=10; b=20;
if (a>15) && (b==20)…
Т.к. первая часть выражения ложна, то результат тоже ложь, независимо от значения второй части.

& и | - вычисляют выражение до конца. Поэтому их нужно использовать только для побитовых операций.

Условный оператор if

if (<логическое выражение>)
    <оператор1>;
[else
    <оператор2>;]
Алгоритм выполнения:
1. Вычисление значения логического выражения
2. Если оно истинно, то выполняется оператор после логического выражения
3. Если оно ложно и в операторе есть else, то выполняется оператор после else

Примеры
1) if (x> 10)
        y=15;

2) if ((x<7) || (x>1000))
        y=15;
else
        y=20;

3) x=7;
y=20;
if (x<3)
        x=-5;
y=12; // не зависит от if
Результат: x=7 y=12

```
4) x=7;
y=20;
if (x<3) {
        x=-5;
        y=12; // зависит от if, т.к. стоят фигурные скобки.
}
```
Результат: x=7 y=20

Тернарный (троичный) условный оператор

(условие) ? выражение1 : выражение2

Если условие истинно, то возвращается выражение1, иначе выражение2.

Выражение

```
if ((x>=10) && (x<=20))
        y=0;
else
        y=8;
```

можно записать компактнее:

```
y = ((x>=10) && (x<=20)) ? 0 : 8;
```

Оператор выбора switch

```
switch ( <выражение> ) {
        case <константа1>:
                <операторы>;
        case <константа2>:
                <операторы>;
        . . .
        [default:
                <операторы>;]
}
```

Значение выражения последовательно сравнивается с константами, указанными после case. Если выражение совпадает с константой некоторого блока, то выполняется оператор этого блока. При наличии в блоке оператора break выполняется выход из оператора выбора. Если выражение не совпадает ни с одной из констант, выполняется раздел default.

Выражение и константы должны быть целочисленными или строковыми.

При записи вариантов выбора могут использоваться только константы и константные выражения (использовать переменные нельзя).

Пример 1: по номеру дня недели напечатать название

```
switch (day) {
        case 1:   System.out.println("понедельник"); break;
        case 2:   System.out.println("вторник"); break;
        case 3:   System.out.println("среда"); break;
        …
        case 7:   System.out.println("воскресенье"); break;
        default:  System.out.println("ошибка");
}
```

Пример 2:

Выражение, написанное через if, можно переписать через switch.

```
if (x>10)
        x=20;
else
        y=15;

switch ((x>10) ? 1 : 0) {
        case 1: x=20; break;
        case 0: y=15;
}

switch ((x>10) ? 1 : 0) {
        case 1: x=20; break;
        default: y=15;
}
```