



链滴

java 运算符

作者: [ibut](#)

原文链接: <https://ld246.com/article/1524220548247>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

运算符归类:

一元运算符: 逻辑运算符(!), 非运算符(~)

二元运算符: 复制运算符(=), 逻辑与运算符(&&), 逻辑或运算符(||), 逐位与运算符(&), 逐位或运算符(|), 右移运算符(>>)

三元运算符: X?Y:Z;

逐位运算符

逐位运算符是一种比较复制的运算符。可以分为逐位逻辑运算符, 逐位位移运算符两种。无论那种逐运算符, 都必须将操作数要求是整形的操作符, 转换为32位的二进制数值。然后参与运算, 运算完毕结果转换成十进制数据。

1. 逐位与运算符

逐位与运算符(&)是一个二元运算符, 该运算符可以将左右两个操作数据逐位执行And的操作

即只有2个操作数据中的对应的位数都加1是, 该结果中的的这一位才是1, 否则就是0, 如下代码:

```
int s=9&12; //结果是8
int s1=1&15; //结果是1
```

十进制	二进制	
9	0000 0000 0000 0000 0000 0000 0000 1001	
12	0000 0000 0000 0000 0000 0000 0000 1100	
----- 逐位与操作		
8	0000 0000 0000 0000 0000 0000 0000 1000	
十进制	二进制	
1	0000 0000 0000 0000 0000 0000 0000 0001	
15	0000 0000 0000 0000 0000 0000 0000 1111	
----- 逐位与操作		
1	0000 0000 0000 0000 0000 0000 0000 0001	

图解: 在进行逐位与操作时, 逐位与运算符会将10进制的数据转换成2进制的数据, 在将2进制的数据中的每一位数值逐位进行and操作, 的出结果后将结果转换成10进制的数据。

2. 逐位或运算符

逐位或运算符(|)和逐位与运算符类似, 可以将左右两个操作符逐位都执行OR操作, 两个操作中的相位的只要有一个为1的时, 该结果的这一位就是1, 其他就是0。

3. 逐位异或运算符

逐位异或运算符(^)和逐位与运算符类似, 可以将左右2个操作数逐位执行异或运算, 所谓异或操作是, 将第1个操作数和第2个操作数相对应的位上的2个数值异或, 不同为1, 相同为0。

4. 逐位非运算符

逐位非运算符(~)是一元运算符, 可以将操作数据中的所有位数中的数值**取反**, 在java中对于一个带符的整数进行逐位非运算符。相当于将该整数改变符号, 然后在减1。

5. 左移运算符

左移运算符(<<)是一个二元运算符, 可以将1个操作数据中的所有数值(一共32位), 向左移动, 移动

位数由第2个操作数决定，因此第2个操作数应该是0~31的整数，如果第2个操作数据大于31，那么结果就是第一个数据

十进制	二进制	
12	0000 0000 0000 0000 0000 0000 0000 1100	
-----		左移操作
24	0000 0000 0000 0000 0000 0000 0001 1000	
-----		左移操作
15	0000 0000 0000 0000 0000 0000 0000 1111	
-----		左移操作
122880	0000 0000 0000 0001 1110 0000 0000 0000	

图解：是将一个值左移1位，相当于将该数值乘以2；左移2位，相当于将该数值乘以4。

6带符号的右移运算符

带符号的右移运算符(>>) 是一个二元操作符，可以将第1个操作数据中的所有数值(一共32位)向右移，移动的位数由第2个操作数据来决定，因此第2个操作数据应该是0~31的整数，如果第2个操作数据于31，那么结果就是第一个数据。

7用0补足的右移运算符

用0补足的右移运算符(>>>)与带符号的左移运算符类似，只是在右移的时候，最左侧的数值都是用0补充。

```
int s=12>>>1; //结果是6
int s1=-12>>>1; //结果是2147483642
```