



链滴

java 中的移动位运算：<<,>>,>>>

作者：[Maggie](#)

原文链接：<https://ld246.com/article/1537860390560>

来源网站：[链滴](#)

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

1. 概念

- << 左移运算符，左移是在后面补0， num << 1,相当于num乘以2
- >> 右移运算符，右移是在前面补1或0， num >> 1, 相当于num除以2
- >>> 无符号右移，是在前面补0， 忽略符号位， 空位都以0补齐

另外， 不论是左右还是右移32位， 相当于不移动， 还是原值。

实际上 在java虚拟机执行这句代码的时候如下这样执行的：

5>>(n%32)--->结果

你这里n=32； 所以5>>32即是 5>>(32%32)-->5>>0 的结果；

2. 测试代码

```
public class Test{
    public Test(){
        System.out.println("=====算术右移 >> =====");
        int i=0xC0000000;
        System.out.println("移位前: i= "+i+" = "+Integer.toBinaryString(i)+"(B)");
        i=i>>28;
        System.out.println("移位后: i= "+i+" = "+Integer.toBinaryString(i)+"(B)");

        System.out.println("-----");

        int j=0x0C000000;
        System.out.println("移位前: j= "+j+" = "+Integer.toBinaryString(j)+"(B)");
        j=j>>24;
        System.out.println("移位后: j= "+j+" = "+Integer.toBinaryString(j)+"(B)");

        System.out.println("\n");
        System.out.println("=====逻辑右移 >>> =====");
        int m=0xC0000000;
        System.out.println("移位前: m= "+m+" = "+Integer.toBinaryString(m)+"(B)");
        m=m>>>28;
        System.out.println("移位后: m= "+m+" = "+Integer.toBinaryString(m)+"(B)");

        System.out.println("-----");

        int n=0x0C000000;
        System.out.println("移位前: n= "+n+" = "+Integer.toBinaryString(n)+"(B)");
        n=n>>24;
        System.out.println("移位后: n= "+n+" = "+Integer.toBinaryString(n)+"(B)");

        System.out.println("\n");
        System.out.println("=====移位符号的取模=====");
        int a=0xCC000000;
        System.out.println("移位前: a= "+a+" = "+Integer.toBinaryString(a)+"(B)");
        System.out.println("算术右移32: a="+a>>32+" = "+Integer.toBinaryString(a>>32)+"(B)");
    };
    System.out.println("逻辑右移32: a="+a>>>32+" = "+Integer.toBinaryString(a>>>32)+"(B)");
};
```

```

        System.out.println("算术右移64: a="+a>>64+" = "+Integer.toBinaryString(a>>64)+"(B)");
    };
    System.out.println("逻辑右移64: a="+a>>>64+" = "+Integer.toBinaryString(a>>>64)+"(B)");

    }

    public static void main(String[] args){
        new Test();
    }
}

```

运行结果:

=====算术右移 >> =====
 移位前: j= -1073741824 = 110000000000000000000000
 0000000000(B)
 移位后: i= -4 = 111111111111111111111111111111111110
 (B)

移位前: j= 201326592 = 110000000000000000000000000000(B)
 移位后: j= 12 = 1100(B)

=====逻辑右移 >>> =====
 移位前: m= -1073741824 = 110000000000000000000000
 0000000000(B)
 移位后: m= 12 = 1100(B)

移位前: n= 201326592 = 110000000000000000000000000000(B)
 移位后: n= 12 = 1100(B)

=====移位符号的取模=====
 移位前: a= -872415232 = 110011000000000000000000000000(B)
 算术右移32: a=-872415232 = 110011000000000000000000000000(B)
 逻辑右移32: a=-872415232 = 110011000000000000000000000000(B)
 算术右移64: a=-872415232 = 110011000000000000000000000000(B)
 逻辑右移64: a=-872415232 = 110011000000000000000000000000(B)

3. 为什么没有无符号左移

这个问题大家可以思考一下，应该能想出来。（提示：没有就是没有存在的意思）

作者 @没有故事的老大爷

如果你还没有找到一个睡觉时都还在赚钱的方法，那么你就干到死。

