

JAVA的三个移位运算符计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/642/2021_2022_JAVA_E7_9A_84_E4_B8_89_c97_642834.htm 编辑特别推荐: 全国计算机等级考试（等考）指定教材 全国计算机等级考试学习视频 全国计算机等级考试网上辅导招生 全国计算机等级考试时间及科目预告 百考试题教育全国计算机等级考试在线测试平台 全国计算机等级考试资料下载

gt.(lt.):带符号右（左）移，高位补符号位。 gt.lt.lt.):不带符号右（左）移，高位补零。 lt.:左移，末尾补零。

```
view plaincopy to clipboardprint? public static void
main(String[] args) { int i = 11. String si = Integer.toBinaryString(i).
String si2 = Integer.toBinaryString(igt.gt.lt.gt.gt.结果\t" si2).
System.out.println("正数gt.结果\t" si3). System.out.println("正数lt.
结果\t" si4). //as for nagative number int j = -11. String sj =
Integer.toBinaryString(j). String sj2 =
Integer.toBinaryString(jgt.gt.lt.gt.gt.结果\t" sj2).
System.out.println("负数gt.结果\t" sj3). System.out.println("负数lt.
结果\t" sj4). } public static void main(String[] args) { int i = 11.
String si = Integer.toBinaryString(i). String si2 =
Integer.toBinaryString(igt.gt.lt.gt.gt.结果\t" si2).
System.out.println("正数gt.结果\t" si3). System.out.println("正数lt.
结果\t" si4). //as for nagative number int j = -11. String sj =
Integer.toBinaryString(j). String sj2 =
Integer.toBinaryString(jgt.gt.lt.gt.gt.结果\t" sj2).
System.out.println("负数gt.结果\t" sj3). System.out.println("负数lt.
结果\t" sj4). } view plaincopy to clipboardprint? org: 1011 正
```

数gt.gt.lt.gt.gt.结果 11111111111111111111111111111101 负数gt.结果
1111111111111111111111111111111101 负数lt.结果
11111111111111111111111111111010100 org: 1011 正数gt.gt.lt.gt.gt.结果
11111111111111111111111111111101 负数gt.结果
11111111111111111111111111111101 负数lt.结果
11111111111111111111111111111010100 在Thinking in Java第三章中
的一段话: 移位运算符面向的运算对象也是二进制的“位”。
可单独用它们处理整数类型（主类型的一种）。左移位运算符（lt.）
能将运算符左边的运算对象向左移动运算符右侧指定的位数（在低位补0）。
“有符号”右移位运算符（gt.）则将运算符左边的运算对象向右移动运算符右侧指定的位数。
“有符号”右移位运算符使用了“符号扩展”：若值为正，则在高位插入0；
若值为负，则在高位插入1。Java也添加了一种“无符号”右移位运算符（gt.>.），
它使用了“零扩展”：无论正负，都在高位插入0。这一运算符是C或C++没有的。
若对char，byte或者short进行移位处理，那么在移位进行之前，它们会自动
转换成一个int。只有右侧的5个低位才会用到。这样可防止我们在一个int数里
移动不切实际的位数。若对一个long值进行处理，最后得到的结果也是long。
此时只会用到右侧的6个低位，防止移动超过long值里现成的位数。但在进行
“无符号”右移位时，也可能遇到一个问题。若对byte或short值进行右移位运算，
得到的可能不是正确的结果（Java 1.0和Java 1.1特别突出）。它们会自动转换
成int类型，并进行右移位。但“零扩展”不会发生，所以在那些情况下会得到-1
的结果。??无法理解? 100Test 下载频道开通，各类考试题目直接下载。
详细请访问 www.100test.com