

Java库谜题64：按余数编组 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/138/2021_2022_Java_E5_BA_93_E8_B0_9C_c97_138631.htm 下面的程序将生成整数对3取余的柱状图，那么，它将打印出什么呢？

```
public class Mod {
    public static void main(String[ ] args) { final int MODULUS = 3.
    int[] histogram = new int[MODULUS]. // Iterate over all ints
    (Idiom from Puzzle 26) int i = Integer.MIN_VALUE. do {
    histogram[Math.abs(i) % MODULUS] . } while (i !=
    Integer.MAX_VALUE). for (int j = 0. j
```

```
System.out.println(histogram[j] " "). } }
```

该程序首先初始化int数组histogram，其每一个位置都为对3取余的一个数值而准备（0、1和2），所有这三个位置都被初始化为0。然后，该程序在所有232个int数值上遍历变量i，使用的是在谜题26中介绍的惯用法。因为整数取余操作（%）在第一个操作数是负数时，可以返回一个负值，就像在谜题1中所描述的那样，所以该程序在计算i被3整除的余数之前，先取i的绝对值。然后用这个余数来递增数组位置的索引。在循环完成之后，该程序将打印histogram数组中的内容，它的元素表示对3取余得到0、1和2的int数值的个数。该程序所打印的三个数字应该彼此大致相等，它们加起来应该等于232。如果你想知道怎样计算出它们的精确值，那么你需要有一点数学气质，并仔细阅读下面两段话。否则，你可以跳过这两段话。该程序打印的三个数字不可能精确地相等，因为它们必须加起来等于232，这个数字不能被3除尽。如果你仔细观察2的连续幂级数对3取余的值，就会发现，它们在1和2之间交替变化：20对3取余是1

, 21对3取余是2, 22对3取余是1, 23对3取余是2, 以此类推。每一个2的偶次幂对3取余的值都是1, 每一个2的奇次幂对3取余的值都是2。因为232对3取余是1, 所以该程序所打印的三个数字中有一个将比另外两个大1, 但是它是哪一个呢? 该循环依次递增三个数组元素的数值, 因此该循环最后递增的那个数值必然是最大的数值, 它就是表示Integer.MAX_VALUE或(232-1)对3取余的数值。因为231是2的奇次幂, 所以它对3取余应该得到2, 因此(232-1)对3取余将得到1。该程序打印的三个数字中的第二个表示的就是对3取余得到1的int数值的个数, 因此, 我们期望这个值比第一个和最后一个数值大1。由此, 该程序应该在运行了相当长的时间之后, 打印(232/3)的较小值 (232/3)的较大值 (232/3)的较小值, 即1431655765 1431655766 1431655765。但是它真的是这么做的吗? 不, 它几乎立刻就抛出了下面的异常: Exception in thread "main" ArrayIndexOutOfBoundsException: -2 at Mod.main(Mod.java:9) 问题出在哪了呢? 问题在于该程序对Math.abs方法的使用上, 它会导致错误的对3取余的数值。考虑一下当i为 -2时所发生的事情, 该程序计算 Math.abs(-2) % 3的数值, 得到2, 但是-2对3取余应该得到1。这可以解释为什么产生了不正确的统计结果, 但是还有一个问题留待解决, 为什么程序抛出了ArrayIndexOutOfBoundsException异常呢? 这个异常表明该程序使用了一个负的数组索引, 但是这肯定是不可能的: 数组索引是通过接受i的绝对值并计算这个绝对值被3整除时的余数而计算出来的。在计算一个非负的int数值整除一个正的int数值的余数时, 可以保证将产生一个非负的结果[JLS 15.17.3]。我们又要问了, 这里又出了什么

问题呢？要回答这个问题，我们必须要去看看Math.abs的文档。这个方法的名字有一点带有欺骗性，它几乎总是返回它的参数的绝对值，但是在有一种情况下，它做不到这一点。文档中叙述道：“如果其参数等于Integer.MIN_VALUE，那么产生的结果与该参数相同，它是一个负数。”通过对这条知识的掌握，就可以很清楚地知道为什么该程序立即抛出了ArrayIndexOutOfBoundsException异常。循环索引i的初始值是Integer.MIN_VALUE，由Math.abs(Integer.MIN_VALUE) % 3所产生的数组索引等于Integer.MIN_VALUE % 3，即-2。为了订正这个程序，我们必须用一个真正的取余操作来替代伪取余计算(Math.abs(i) % MODULUS)。如果我们将这个表达式替换为对下面这个方法的调用，那么该程序就可以产生我们做期望的输出1431655765 1431655766 1431655765：100Test 下载

频道开通，各类考试题目直接下载。详细请访问
www.100test.com