

JAVA循环谜题26:在循环中 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\\_BE\\_AA\\_E7\\_8E\\_AF\\_c97\\_138540.htm](https://www.100test.com/kao_ti2020/138/2021_2022_JAVA_E5_BE_AA_E7_8E_AF_c97_138540.htm) 下面的程序计算了一个循环的迭代次数，并且在循环终止时将这个计数值打印了出来。

那么，它打印的是什么呢？  

```
public class InTheLoop { public static final int END = Integer.MAX_VALUE. public static final int START = END - 100. public static void main(String[] args) { int count = 0. for (int i = START. i count . System.out.println(count). } }
```

如果你没有非常仔细地查看这个程序，你可能会认为它将打印100，因为END比START大100。如果你稍微仔细一点，你可能会发现该程序没有使用典型的循环惯用法。大多数的循环会在循环索引小于终止值时持续运行，而这个循环则是在循环索引小于或等于终止值时持续运行。所以它会打印101，对吗？嗯，根本不对。如果你运行该程序，就会发现它压根就什么都没有打印。更糟的是，它会持续运行直到你撤销它为止。它从来都没有机会去打印count，因为在打印它的语句之前插入的是一个无限循环。问题在于这个循环会在循环索引（i）小于或等于Integer.MAX\_VALUE时持续运行，但是所有的int变量都是小于或等于Integer.MAX\_VALUE的。因为它被定义为所有int数值中的最大值。当i达

到Integer.MAX\_VALUE，并且再次被执行增量操作时，它就绕回到了Integer.MIN\_VALUE。如果你需要的循环会迭代到int数值的边界附近时，你最好是使用一个long变量作为循环索引。只需将循环索引的类型从int改变为long就可以解决该问题，从而使程序打印出我们所期望的101：

```
for (long i =
```

START. *i* 更一般地讲，这里的教训就是 `int` 不能表示所有的整数。无论你在何时使用了一个整数类型，都要意识到其边界条件。如果其数值下溢或是上溢了，会怎么样呢？所以通常最好是使用一个取之范围更大的类型。（整数类型包括 `byte`、`char`、`short`、`int` 和 `long`。）不使用 `long` 类型的循环索引变量也可以解决这个问题，但是它看起来并不那么漂亮：`int i = START. do { count. }while (i != END)`。如果清晰性和简洁性占据了极其重要的地位，那么在这种情况下使用一个 `long` 类型的循环索引几乎总是最佳方案。但是有一个例外：如果你在所有的（或者几乎所有的）`int` 数值上迭代，那么使用 `int` 类型的循环索引的速度大约可以提高一倍。下面是将 `f` 函数作用于所有 40 亿个 `int` 数值上的惯用法：`//Apply the function f to all four billion int values int i = Integer.MIN_VALUE. do { f(i). }while (i != Integer.MAX_VALUE)`。该谜题对语言设计者的教训与谜题 3 相同：可能真的值得去考虑，应该对那些不会在产生溢出时而不抛出异常的算术运算提供支持。同时，可能还值得去考虑，应该对那些在整数值范围之上进行迭代的循环进行特殊设计，就像许多其他语言所做的那样。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](http://www.100test.com)