

JAVA循环谜题25:无情的增量操作 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_138544.htm 下面的程序对一个变量重复

地进行增量操作，然后打印它的值。那么这个值是什么呢？

```
public class Increment { public static void main(String[] args) { int j = 0. for (int i = 0. i j = j . System.out.println(j). } }
```

乍一看，这个程序可能会打印100。毕竟，它对j做了100次增量操作。可能会令你感到有些震惊，它打印的不是100而是0。所有的增量操作都无影无踪了，为什么？就像本谜题的题目所暗示的，问题出在了执行增量操作的语句上：`j=j`。大概该语句的作者是想让它执行对j的值加1的操作，也就是表达式j所做的操作。遗憾的是，作者大咧咧地将这个表达式的值有赋回给了j。当操作符被置于一个变量值之后时，其作用就是一个后缀增量操作符（postfix increment operator）[JLS 15.14.2]：表达式j的值等于j在执行增量操作之前的初始值。因此，前面提到的赋值语句首先保存j的值，然后将j设置为其值加1，最后将j复位到它的初始值。换句话说，这个赋值操作等价于下面的语句序列：`int tmp = j. j = j + 1. j = tmp?`。程序重复该过程100次，之后j的值还是等于它在循环开始之前的值，即0。订正该程序非常简单，只需从循环中移除无关的赋值操作，只留下：`for (int i = 0. i j.` 经过这样的修改，程序就可以打印出我们所期望的100了。这与谜题7中的教训相同：不要在单个的表达式中对相同的变量赋值超过一次。对相同的变量进行多次赋值的表达式会产生混淆，并且很少能够产生你希望的行为。

100Test 下载频道开通，各类考试题目直接下载。详细请访问

