

JAVA循环谜题30:循环者的爱子 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\\_138530.htm](https://www.100test.com/kao_ti2020/138/2021_2022_JAVA_E5_BE_AA_E7_8E_AF_c97_138530.htm) 请提供一个对*i*的声明，将下面的循环转变为一个无限循环：`while (i != i 0) {}` 与前一个谜题不同，你必须在你的答案中不使用浮点数。换句话说，你不能把*i*声明为double或float类型的。与前一个谜题一样，这个谜题初看起来是不可能实现的。毕竟，一个数字总是等于它自身加上0，你被禁止使用浮点数，因此不能使用NaN，而在整数类型中没有NaN的等价物。那么，你能给出什么呢？我们必然可以得出这样的结论，即*i*的类型必须是非数值类型的，并且这其中存在着解谜方案。唯一的操作符有定义的非数值类型就是String。操作符被重载了：对于String类型，它执行的不是加法而是字符串连接。如果在连接中的某个操作数具有非String的类型，那么这个操作数就会在连接之前转换成字符串[JLS 15.18.1]。事实上，*i*可以被初始化为任何值，只要它是String类型的即可，例如：`String i = "Buy seventeen copies of Effective Java"`。int类型的数值0被转换成String类型的数值"0"，并且被追加到了感叹号之后，所产生的字符串在使用equals方法计算时就不等于最初的字符串了，这样它们在使用==操作符进行计算时，当然就不是相等的。因此，计算布尔表达式*(i != i 0)*得到的值就是true，循环也就永远不会被终止了。总之，操作符重载是很容易令人误解的。在本谜题中的加号看起来是表示一个加法，但是通过为变量*i*选择合适的类型，即String，我们让它执行了字符串连接操作。甚至是因为变量被命名为*i*，都使得本谜题更加容易令人误解，因为*i*通

常被当作整型变量名而被保留的。对于程序的可读性来说，好的变量名、方法名和类名至少与好的注释同等重要。对语言设计者的教训与谜题11和13中的教训相同。操作符重载是很容易引起混乱的，也许操作符就不应该被重载用来进行字符串连接操作。有充分的理由证明提供一个字符串连接操作符是多么必要，但是它不应该是。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)