

JAVA异常谜题45:令人疲惫不堪的测验 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\\_BC\\_82\\_E5\\_B8\\_B8\\_c97\\_138495.htm](https://www.100test.com/kao_ti2020/138/2021_2022_JAVA_E5_BC_82_E5_B8_B8_c97_138495.htm) 本谜题将测试你对递归的了解程度。下面的程序将做些什么呢？

```
public class Workout {  
    public static void main(String[] args) { workHard().  
        System.out.println("It ' s nap time."). } private static void  
workHard() { try { workHard(). } finally { workHard(). } } }
```

要不是有try-finally语句，该程序的行为将非常明显：workHard方法递归地调用它自身，直到程序抛出StackOverflowError，在此刻它以这个未捕获的异常而终止。但是，try-finally语句把事情搞得复杂了。当它试图抛出StackOverflowError时，程序将会在finally语句块的workHard方法中终止，这样，它就递归调用了自己。这看起来确实就像是一个无限循环的秘方，但是这个程序真的会无限循环下去吗？如果你运行它，它似乎确实是这么做的，但是要想确认的唯一方式就是分析它的行为。Java虚拟机对栈的深度限制到了某个预设的水平。当超过这个水平时，VM就抛出StackOverflowError。为了让我们能够更方便地考虑程序的行为，我们假设栈的深度为3，这比它实际的深度要小得多。现在让我们来跟踪其执行过程。main方法调用workHard，而它又从其try语句块中递归地调用了自己，然后它再一次从其try语句块中调用了自己。在此时，栈的深度是3。当workHard方法试图从其try语句块中再次调用自己时，该调用立即就会以StackOverflowError而失败。这个错误是在最内部的finally语句块中被捕获的，在此处栈的深度已经达到了3。在那里，workHard方法试图递归地调用它自己，但

是该调用却以StackOverflowError而失败。这个错误将在上一级的finally语句块中被捕获，在此处站的深度是2。该finally中的调用将与相对应的try语句块具有相同的行为：最终都会产生一个StackOverflowError。这似乎形成了一种模式，而事实也确实如此。WorkOut的运行过程如左面的图所示。在这张图中，对workHard的调用用箭头表示，workHard的执行用圆圈表示。所有的调用除了一个之外，都是递归的。会立即产生StackOverflowError异常的调用用由灰色圆圈前导的箭头表示，try语句块中的调用用向左边的向下箭头表示，finally语句块中的调用用向右边的向下箭头表示。箭头上的数字描述了调用的顺序。这张图展示了一个深度为0的调用（即main中的调用），两个深度为1的调用，四个深度为2的调用，和八个深度为3的调用，总共是15个调用。那八个深度为3的调用每一个都会立即产生StackOverflowError。至少在把栈的深度限制为3的VM上，该程序不会是一个无限循环：它在15个调用和8个异常之后就会终止。但是对于真实的VM又会怎样呢？它仍然不会是一个无限循环。其调用图与前面的图相似，只不过要大得多得多而已。那么，究竟大到什么程度呢？有一个快速的试验表明许多VM都将栈的深度限制为1024，因此，调用的数量就是 $1\ 2\ 4\ 8\ \dots\ 21,024=21,025-1$ ，而抛出的异常的数量是21,024。假设我们的机器可以在每秒钟内执行 $10^{10}$ 个调用，并产生 $10^{10}$ 个异常，按照当前的标准，这个假设的数量已经相当高了。在这样的假设条件下，程序将在大约 $1.7 \times 10^{291}$ 年后终止。为了让你对这个时间有直观的概念，我告诉你，我们的太阳的生命周期大约是1010年，所以我们可以很确定，我们中没有任何人能够看到这个程序终止的时刻。

尽管它不是一个无限循环，但是它也就算是一个无限循环吧。从技术角度讲，调用图是一棵完全二叉树，它的深度就是VM的栈深度的上限。WorkOut程序的执行过程等于是在先序遍历这棵树。在先序遍历中，程序先访问一个节点，然后递归地访问它的左子树和右子树。对于树中的每一条边，都会产生一个调用，而对于树中的每一个节点，都会抛出一个异常。本谜题没有很多关于教训方面的东西。它证明了指数算法对于除了最小输入之外的所有情况都是不可行的，它还表明了你甚至可以不费什么劲就可以编写出一个指数算法。

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)