

Java更多的类谜题67：对字符串上瘾 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\\_E6\\_9B\\_B4\\_E5\\_A4\\_9A\\_c97\\_138623.htm](https://www.100test.com/kao_ti2020/138/2021_2022_Java_E6_9B_B4_E5_A4_9A_c97_138623.htm) 一个名字可以被用来引用位于不同包内的多个类。下面的程序就是在探究当你重用了—个平台类的名字时，会发生什么。你认为它会做些什么呢？尽管这个程序属于那种让你通常一看到就会感到尴尬的程序，但是你还是应该继续下去，把门锁上，把百叶窗拉上，然后试试看：

```
public class StrungOut { public static void main(String[] args) { String s = new String("Hello world"); System.out.println(s); } } class String { private final java.lang.String s; public String(java.lang.String s) { this.s = s; } public java.lang.String toString() { return s; } }
```

如果说这个程序有点让人讨厌的话，它看起来还是相当简单的。在未命名包中的String类就是一个java.lang.String实例的包装器，看起来该程序应该打印Hello world。如果你尝试着运行该程序，你会发现你运行不了它，VM将弹出了一个像下面这样的错误消息：Exception in thread "main" java.lang.NoSuchMethodError: main但是它肯定是一个main方法的：它就白纸黑字地写在那里。为什么VM找不到它呢？VM不能找到main方法是因为它并不在那里。尽管StrungOut有一个被命名为main的方法，但是它却具有错误的签名。一个main方法必须接受一个单一的字符串数组参数[JVMS 5.2]。VM努力要告诉我们的是StrungOut.main接受的是由我们的String类所构成的数组，它无论如何都与java.lang.String没有任何关系。如果你确实需要编写自己的字符串类，看在老天爷的份上，千万不要称其为String。要避

免重用平台类的名字，并且千万不要重用java.lang中的类名，因为这些名字会被各处的程序自动加载。程序员习惯于看到这些名字以无限定的形式出现，并且会很自然地认为这些名字引用的是我们所熟知的java.lang中的类。如果你重用了这些名字的某一个，那么当这个名字在其自己的包内被使用时，该名字的无限定形式将会引用到新的定义上。要订正该程序，只需为这个非标准的字符串类挑选一个合理的名字即可。该程序下面的这个版本很明显是正确的，而且它比最初的版本要更易于理解。它将打印出如你所期望的Hello World:

```
public class StrungOut { public static void main(String[ ] args) {  
MyString s = new MyString("Hello world"). System.out.println(s). }  
} class MyString { private final java.lang.String s. public
```

```
MyString(java.lang.String s) { this.s = s.} public java.lang.String  
toString() { return s.} } 宽泛地讲，本谜题的教训就是要避免重用  
类名，尤其是Java平台类的类名。千万不要重用java.lang包  
内的类名，相同的教训也适用于类库的设计者。Java平台的设计  
者已经在这个问题上栽过数次了，著名的例子
```

有java.sql.Date，它与java.util.Date和org.omg.CORBA.Object相冲突。与在本章中的许多其他谜题一样，这个教训是有关你在除了覆写之外的其他情况应该避免名字重用这一原则的一个具体实例。对平台实现者来说，其教训是诊断信息应该清晰地解释失败的原因。VM应该可以很容易地将没有任何具有正确签名的main方法的情况与根本就没有任何main方法的情况区分开。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)