

JAVA字符谜题11:我的类是什么II 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_AD_97_E7_AC_A6_c97_138552.htm 下面的程序所要做的事情正是前一个谜题所做的事情，但是它没有假设斜杠符号就是分隔文件名组成部分的符号。相反，该程序使用的

是`java.io.File.separator`，它被指定为一个公共的String域，包含了平台相关的文件名分隔符。那么，这个程序会打印出其正确的、平台相关的类文件名吗？

```
package com.javapuzzlers.  
import java.io.File.  
public class MeToo {  
    public static void  
    main(String[] args){  
        System.out.println(MeToo.class.getName().  
            replaceAll("\\.", File.separator) ".class").  
    }  
}
```

这个程序根据底层平台的不同会显示两种行为中的一种。如果文件分隔符是斜杠，就像在UNIX上一样，那么该程序将打

印`com/javapuzzlers/MeToo.class`，这是正确的。但是，如果文件分隔符是反斜杠，就像在Windows上一样，那么该程序将打印像下面这样的内容：

```
Exception in thread "main"  
java.lang.StringIndexOutOfBoundsException: String index out of  
range: 1 at java.lang.String.charAt(String.java:558) at  
java.util.regex.Matcher.appendReplacement(Matcher.java:696) at  
java.util.regex.Matcher.replaceAll(Matcher.java:806) at  
java.lang.String.replaceAll(String.java:2000) at  
com.javapuzzlers.MeToo.main(MeToo.java:6)
```

尽管这种行为是平台相关的，但是它并非就是我们所期待的。在Windows上出了什么错呢？事实证明，`String.replaceAll`的第二个参数不是一个普通的字符串，而是一个替代字符串（replacement string）

，就像在java.util.regex规范中所定义的那样[Java-API]。在替代字符串中出现的反斜杠会把紧随其后的字符进行转义，从而导致其被按字面含义而处理了。当你在Windows上运行该程序时，替代字符串是单独的一个反斜杠，它是无效的。不可否认，抛出的异常应该提供更多一些有用的信息。那么你应该怎样解决此问题呢？5.0版本提供了不是一个而是两个新的方法来解决它。第一个方法

是java.util.regex.Matcher.quoteReplacement，它将字符串转换成相应的替代字符串。下面展示了如何使用这个方法来订正该程序：`System.out.println(MeToo.class.getName().replaceAll("\\\\.", Matcher.quoteReplacement(File.separator)) ".class")`。引入到5.0版本中的第二个方法提供了一个更好的解决方案。该方法就是String.replace(CharSequence, CharSequence)，它做的事情和String.replaceAll相同，但是它将模式和替代物都当作字面含义的字符串处理。下面展示了如何使用这个方法来订正该程序：`System.out.println(MeToo.class.getName().replace(".", File.separator) ".class")`。但是如果你使用的是较早版本的Java该怎么办？很遗憾，没有任何捷径能够生成替代字符串。完全不使用正则表达式，而使用String.replace(char, char)也许要显得更容易一些：`System.out.println(MeToo.class.getName().replace('.', File.separatorChar) ".class")`。本谜题和前一个谜题的主要教训是：在使用不熟悉的类库方法时一定要格外小心。当你心存疑虑时，就要求助于Javadoc。还有就是正则表达式是很棘手的：它所引发的问题趋向于在运行时刻而不是在编译时刻暴露出来。对API的设计者来说，使用方法具名的模式来以明显的方式区分方法行为的差异是很重要的。Java

的String类就没有很好地遵从这一原则。对许多程序员来说，对于哪些字符串替代方法使用的是字面含义的字符串，以及哪些使用的是正则表达式或替代字符串，要记住这些都不是一件容易事。谜题22：URL的愚弄 本谜题利用了Java编程语言中一个很少被人了解的特性。请考虑下面的程序将会做些什么？

```
public class BrowserTest { public static void main(String[] args) { System.out.print("iexplore:"). http://www.google.com. System.out.println(":maximize"). } }
```

这是一个有点诡异的问题。该程序将不会做任何特殊的事情，而是直接打印iexplore::maximize。在程序中间出现的URL是一个语句标号（statement label）[JLS 14.7]后面跟着一行行尾注释（end-of-line comment）[JLS 3.7]。在Java中很少需要标号，这多亏了Java没有goto语句。在本谜题中所引用的“Java编程语言中很少被人了解的特性”实际上就是你可以在任何语句前面放置标号。这个程序标注了一个表达式语句，它是合法的，但是却没什么用处。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