

JAVA字符谜题6:行打印程序 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_138380.htm 行分隔符 (line separator) 是为用来分隔文本行的字符或字符组合而起的名字，并且它在不同的平台上是存在差异的。在Windows平台上，它是CR字符 (回车) 和紧随其后的LF字符 (换行) 组成的，而在UNIX平台上，通常单独的LF字符被当作换行字符来引用。下面的程序将这个字符传递给了println方法，那么，它将打印出什么呢？它的行为是否是依赖于平台的呢？

```
public class  
LinePrinter{ public static void main(String[] args){ // Note: \u000A  
is Unicode representation of linefeed (LF) char c = 0x000A.  
System.out.println(c). } }
```

这个程序的行为是平台无关的：它在任何平台上都不能通过编译。如果你尝试着去编译它，就会得到类似下面的出错信息：LinePrinter.java:3: ' .' expected // Note: \u000A is Unicode representation of linefeed (LF) ^ 1 error 如果你和大多数人一样，那么这条信息对界定问题是毫无用处的。这个谜题的关键就是程序第三行的注释。与最好的注释一样，这条注释也是一种准确的表达，遗憾的是，它有一点准确得过头了。编译器不仅会在将程序解析成为符号之前把Unicode转义字符转换成它们所表示的字符 (谜题14)，而且它是在丢弃注释和空格之前做这些事的[JLS 3.2]。这个程序包含了一个Unicode转移字符 (\u000A)，它位于程序唯一的注释行中。就像注释所陈述的，这个转义字符表示换行符，编译器将在丢弃注释之前适时地转换它。遗憾的是，这个换行符是表示注释开始的两个斜杠符之后的第一个行终结符

(line terminator) ，因此它将终结该注释[JLS 3.4]。所以，该转义字符之后的字 (is Unicode representation of linefeed (LF)) 就不是注释的一部分了，而它们在语法上也不是有效的。订正该程序的最简单的方式就是在注释中移除Unicode转义字符，但是更好的方式是用一个转义字符序列而不是一个十六进制整型字面常量来初始化c，从而消除使用注释的必要：

```
public class LinePrinter{ public static void main(String[] args){ char c = ' \n ' . System.out.println(c). } }
```

只要这么做了，程序就可以编译并运行，但是这仍然是一个有问题的程序：它是平台相关的，这正是本谜题所要表达的真正意图。在某些平台上，例如UNIX，它将打印出两个完整的行分隔符；但是在其它一些平台上，例如Windows，它就不会产生这样的行为。尽管这些输出用肉眼看起来是一样的，但是如果它们要被存储到文件中，或是输出到后续的其它处理程序中，那就很容易引发问题。如果你想打印两行空行，你应该调用println两次。如果使用的是JDK 5.0，那么你可以用带有格式化字符串"%n%n"的printf来代替println。%n的每一次出现都将导致printf打印一个恰当的、与平台相关的行分隔符。我们希望，上面三个谜题已经使你信服：Unicode转义字符绝对会产生混乱。教训很简单：除非确实是必需的，否则就不要使用Unicode转义字符。它们很少是必需的。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