

Java类谜题54：Null与Void 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_E7_B1_BB_E8_B0_9C_c97_138479.htm 下面仍然是经典的Hello World程序的另一个变种。那么，这个变种将打印什么呢？

```
public class Null { public static void greet() { System.out.println("Hello world!"); } public static void main(String[] args) { ((Null) null).greet(). } }
```

这个程序看起来似乎应该抛

出NullPointerException异常，因为其main方法是在常量null上调用greet方法，而你是不可以在null上调用方法的，对吗？嗯，某些时候是可以的。如果你运行该程序，就会发现它打印出了“Hello World!”理解本谜题的关键是Null.greet是一个静态方法。正如你在谜题48中所看到的，在静态方法的调用中，使用表达式作为其限定符并非是一个好主意，而这也正是问题之所在。不仅表达式的值所引用的对象的运行期类型在确定哪一个方法将被调用时并不起任何作用，而且如果对象有标识的话，其标识也不起任何作用。在本例中，没有任何对象，但是这并不会造成任何区别。静态方法调用的限定表达式是可以计算的，但是它的值将被忽略。没有任何要求其值为非空的限制。要想消除该程序中的混乱，你可以用它的类作为限定符来调用greet方法：

```
public static void main(String[] args) { Null.greet(). }
```

然而更好的方式是完全消除限定符：

```
public static void main(String[] args) { greet(). }
```

总之，本谜题的教训与谜题48的完全相同：要么用某种类型来限定静态方法调用，要么就压根不要限定它们。对语言设计者来说，应该不允许用表达式来污染静态方法调用的可能性存在，因为它

们只会产生混乱。 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