

JAVA更多的类谜题74：同一性的危机 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_138568.htm 下面的程序是不完整的，它缺乏对Enigma的声明，这个类扩展自java.lang.Object。请

为Enigma提供一个声明，它可以使该程序打印false：

```
public class Conundrum { public static void main(String[] args) { Enigma e = new Enigma(). System.out.println(e.equals(e)). } }
```

噢，还有一件事：你不能覆写equals方法。乍一看，这似乎不可能实现。因为Object.equals方法将测试对象的同一性，通过Enigma传递给equals方法的对象肯定是与其自身相同的。如果你不能覆写Object.equals方法，那么main方法必然打印true，对吗？别那么快下结论，伙计。尽管本谜题禁止你覆写（override）

Object.equals方法，但是你是可以重载（overload）它的，这也就引出了下面的解谜方案：

```
final class Enigma { // Don ' t do this! public Boolean equals(Enigma other){ return false. } }
```

尽管这个声明能够解决本谜题，但是它的做法确实非常不好的。它违反了谜题58的建议：如果同一个方法的两个重载版本都可以应用于某些参数，那么它们应该具有相同的行为。在本例中，`e.equals(e)`和`e.equals((Object)e)`将返回不同的结果，其潜在的混乱是显而易见的。然而，有一种解谜方案是不会违反这项建议的：

```
final class Enigma { public Enigma() { System.out.println(false). System.exit(0). } }
```

可能会有些争论，这个解谜方案似乎违背了本谜题的精神：能够产生我们想要的输出的println调用出现在了构造器中，而不是在main方法中。然而，它确实解决了这个谜题，你不得不承认它很伶俐。这

里的教训，可以参阅前面的8个谜题和谜题58。如果你重载了一个方法，那么一定要确保所有的重载版本行为一致。

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