

Java库谜题58：产生它的散列码 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_BA_93_E8_B0_9C_c97_138646.htm 本谜题试图从前一个谜题中吸取教训。下面的程序还是由一个Name类和一个main方法构成，这个main方法还是将一个名字放置到一个散列集合中，然后检查该集合是否包含了这个名字。然而，这一次Name类已经覆写了hashCode方法。那么下面的程序将打印出什么呢？

```
import java.util.*. public class Name { private String first, last. public
Name(String first, String last) { this.first = first. this.last = last. }
public boolean equals(Name n) { return n.first.equals(first) amp.
n.last.equals(last). } public int hashCode() { return 31 *
first.hashCode() last.hashCode(). } public static void main(String[ ]
args) { Set s = new HashSet(). s.add(new Name("Donald", "Duck")).
System.out.println( s.contains(new Name("Donald", "Duck"))). } }
```

与谜题57一样，该程序的main方法创建了两个Name实例，它们表示的是相同的名字。这一次使用的名字是Donald Duck而不是Mickey Mouse，但是它们不应该有很大的区别。main方法同样还是将第一个实例置于一个散列集合中，然后检查该集合中是否包含了第二个实例。这一次hashCode方法明显是正确的，因此看起来该程序应该打印true。但是，表象再次欺骗了我们：它总是打印出false。这一次又是哪里出错了呢？

这个程序的缺陷与谜题57中的缺陷很相似，在谜题57中

，Name覆写了equals方法，但是没有覆写hashCode方法；而在本谜题中，Name覆写了hashCode方法，但是没有覆写equals方法。这并不是说Name没有声明一个equals方法，它

确实声明了，但是那是个错误的声明。Name类声明了一个参数类型是Name而不是Object的equals方法。这个类的作者可能想要覆写equals方法，但是却错误地重载了它[JLS 8.4.8.1, 8.4.9]。HashSet类是使用equals(Object)方法来测试元素的相等性的；Name类中声明一个equals(Name)方法对HashSet不造成任何影响。那么Name是从哪里得到了它的equals(Object)方法的呢？它是从Object哪里继承而来的。这个方法只有在它的参数与其上调用该方法的对象完全相同时才返回true。我们的程序中的main方法将一个Name实例插入到了散列集合中，并且测试另一个实例是否存在于该散列集合中，由此可知该测试一定是返回false的。对我们而言，两个实例可以代表那令人惊奇的水禽唐老鸭，但是对散列映射表而言，它们只是两个不相等的对象。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