

Java库谜题57：名字里有什么 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_138644.htm 下面的程序包含了一个简单的不可变类，它表示一个名字，其main方法将一个名字置于一个集合中，并检查该集合是否确实包含了该名字。那么，这个程序到底会打印出什么呢？

```
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(Object o) { if (!(o instanceof Name)) return false; Name n = (Name)o; return n.first.equals(first) & n.last.equals(last); } public static void main(String[] args) { Set s = new HashSet(); s.add(new Name("Mickey", "Mouse")); System.out.println( s.contains(new Name("Mickey", "Mouse"))); } }
```

一个Name实例由一个姓和一个名构成。两个Name实例在通过equals方法进行计算时，如果它们的姓相等且名也相等，则这两个Name实例相等。姓和名是用在String中定义的equals方法来比较的，两个字符串如果以相同的顺序包含相同的若干个字符，那么它们就相等。因此，两个Name实例如果表示相同的名字，那么它们就相等。例如，下面的方法调用将返回true：`new Name("Mickey", "Mouse").equals(new Name("Mickey", "Mouse"))` 该程序的main方法创建了两个Name实例，它们都表示Mickey Mouse。该程序将第一个实例放置到了一个散列集合中，然后检查该集合是否包含第二个实例。这两个Name实例是相等的，因此看起来该程序似乎应该打印true。如果你运行它，几乎可以肯定它将打印false。那么这个程序出了什么问题呢？这里的bug在

于Name违反了hashCode约定。这看起来有点奇怪，因为Name连hashCode都没有，但是这确实是问题所在。Name类覆写了equals方法，而hashCode约定要求相等的对象要具有相同的散列码。为了遵守这项约定，无论何时，只要你覆写了equals方法，你就必须同时覆写hashCode方法[EJ Item 8]。因为Name类没有覆写hashCode方法，所以它从Object那里继承了其hashCode实现。这个实现返回的是基于标识的散列码。换句话说，不同的对象几乎总是产生不相等的散列值，即使它们是相等的也是如此。所以说Name没有遵守hashCode的约定，因此包含Name元素的散列集合的行为是不确定的。当程序将第一个Name实例放置到散列集合中时，该集合就会在某个散列位置上放置这个实例对应的项。该集合是基于实例的散列值来选择散列位置的，这个散列值是通过实例的hashCode方法计算出来的。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