

Java更多的库谜题83：诵读困难者的一神论 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_138705.htm 从前有一个人，他认为世上只有一只不寻常的狗，所以他写出了如下的类，将它作为一个单件（singleton）[Gamma95]：

```
public class Dog extends Exception { public static final Dog INSTANCE = new Dog(). private Dog() {} public String toString(){ return "Woof". } }
```

结果证明这个人的做法是错误的。你能够在这个类的外部不使用反射来创建出第2个Dog实例吗？这个类可能看起来像一个单件，但它并不是。问题在于，Dog扩展了Exception，而Exception实现了java.io.Serializable。这就意味着Dog是可序列化的

（serializable），并且解序列（deserialization）会创建一个隐藏的构造器。正如下面的这段程序所演示的，如果你序列化了Dog.INSTANCE，然后对得到的字节序列（byte sequence）进行解序列，最后你就会得到另外一个Dog。该程序打印的是false，表示新的Dog实例和原来的那个实例是不同的，并且它还打印了Woof，说明新的Dog实例也具有相应的功能：

```
import java.io.*. public class CopyDog{ // Not to be confused with copycat public static void main(String[] args){ Dog newDog = (Dog) deepCopy(Dog.INSTANCE). System.out.println(newDog == Dog.INSTANCE). System.out.println(newDog). } // This method is very slow and generally a bad idea! static public Object deepCopy(Object obj){ try{ ByteArrayOutputStream bos = new ByteArrayOutputStream(). new ObjectOutputStream(bos).writeObject(obj). ByteArrayInputStream
```

```
bin = new ByteArrayInputStream(bos.toByteArray()). return new  
ObjectInputStream(bin).readObject(). } catch(Exception e) { throw  
new IllegalArgumentException(e). } } } 要订正这个问题，可  
在Dog中添加一个readResolve方法，它可以那个隐藏的构造  
器转变为一个隐藏的静态工厂（static factory），以返回原来  
那个的Dog [EJ Items 2,57]。在Dog中添加了这个方法之后  
，CopyDog将打印true而不是false，表示那个“复本”实际上  
就是原来的那个实例：private Object readResolve(){ // Accept  
no substitutes! return INSTANCE. } 这个谜题的主要教训就是一个  
实现了Serializable的单件类，必须有一个readResolve方法，  
用以返回它的唯一的实例。一个次要的教训就是，有可能由  
于对一个实现了Serializable的类进行了扩展，或者由于实现  
了一个扩展自Serializable的接口，使得我们在无意中实现  
了Serializable。给平台设计者的教训是，隐藏的构造器，例如  
序列化中产生的那个，会让读者对程序行为的产生错觉。
```

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