

JAVA更多的类谜题73：你的隐私正在公开 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_138570.htm 私有成员，即私有方法、域和类型这些概念的幕后思想是它们只是实现细节：一个类的实现者可以随意地添加一个新的私有成员，或者修改和移除一个旧的私有成员，而不需要担心对该类的客户造成任何损害。换句话说，私有成员被包含它们的类完全封装了。遗憾的是，在这种严密的盔甲保护中仍然存在细小的裂缝。例如，序列化就可以打破这种封装。如果使一个类成为可序列化的，并且接受缺省的序列化形式，那么该类的私有实例域将成为其导出API的一部分[EJ Item 54,55]。当客户正在使用现有的被序列化对象时，对私有表示的修改将会导致异常或者是错误的行为。但是编译期的错误又会怎么样呢？你能否写出一个final的“库”类和“客户”类，这两者都可以毫无问题地通过编译，然后在库类中添加一个私有成员，使得库类仍然能够编译，而客户类却再也不能编译了？如果你的解谜方案是要对库类添加一个私有构造器，以抑制通过缺省的公共构造器而创建实例的行为，那么你只是一知半解。本谜题要求你添加一个私有成员，严格地讲，构造器不是成员[JLS 6.4.3]。本谜题有数个解谜方案，其中一个是使用遮蔽：

```
package library. public final class Api { // private static class String{ } public static String newString() { return new String(). } } package client. import library.Api. public class Client { String s = Api.newString(). }
```

如上编写，该程序就可以毫无问题地通过编译。如果我们不注释掉library.Api中的局部类String的私有声明，那

么Api.newString方法就再也不会返回java.lang.String类型了，因此变量Client.s的初始化将不能通过编译：client/Client.java:4: incompatible types found: library.Api.String, required:

```
java.lang.String String s = Api.newString().
```

^ 尽管我们所做的文本修改仅仅是添加了一个私有类声明，但是我们间接地修改了一个现有公共方法的返回类型，而这是一个不兼容的API修改，因为我们修改了一个被导出API所使用的名字的含义。

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