

Java更多的类谜题66：一件私事 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_138624.htm 在下面的程序中，子类的一个域具有与超类的一个域相同的名字。那么，这个程序会打印出什么呢？

```
class Base { public String className = "Base". } class Derived extends Base { private String className = "Derived". } public class PrivateMatter { public static void main(String[ ] args) { System.out.println(new Derived().className). } }
```

对该程序的表面分析可能会认为它应该打印Derived，因为这正是存储在每一个Derived实例的className域中的内容。更深入一点的分析会认为Derived类不能编译，因为Derived中的className变量具有比Base中的className变量更具限制性的访问权限。如果你尝试着编译该程序，就会发现这种分析也不正确。该程序确实不能编译，但是错误却出在PrivateMatter中。如果className是一个实例方法，而不是一个实例域，那么Derived.className()将覆写Base.className()，而这样的程序是非法的。一个覆写方法的访问修饰符所提供的访问权限与被覆写方法的访问修饰符所提供的访问权限相比，至少要一样多[JLS 8.4.8.3]。因为className是一个域，所以Derived.className隐藏（hide）了Base.className，而不是覆盖了它[JLS 8.3]。对一个域来说，当它要隐藏另一个域时，如果隐藏域的访问修饰符提供的访问权限比被隐藏域的少，尽管这么做不可取的，但是它确实是合法的。事实上，对于隐藏域来说，如果它具有与被隐藏域完全无关的类型，也是合法的：即使Derived.className是GregorianCalendar类型的，Derived类也是合法的。在我们

的程序中的编译错误出现在PrivateMatter类试图访问Derived.className的时候。尽管Base有一个公共域className，但是这个域没有被继承到Derived类中，因为它被Derived.className隐藏了。在Derived类内部，域名className引用的是私有域Derived.className。因为这个域被声明为是private的，所以它对于PrivateMatter来说是不可访问的。因此，编译器产生了类似下面这样的一条错误信息：

```
PrivateMatter.java:11: className has private access in Derived
System.out.println(new Derived().className). ^ 请注意，尽管在Derived实例中的公共域Base.className被隐藏了，但是我们还是可以通过将Derived实例转型为Base来访问到它。下面版本的PrivateMatter就可以打印出Base： public class PrivateMatter { public static void main(String[] args) {
```

```
System.out.println(((Base)new Derived()).className). } } 这说明了覆写与隐藏之间的一个非常大的区别。一旦一个方法在子类中被覆写，你就不能在子类的实例上调用它了（除了在子类内部，通过使用super关键字来方法）。然而，你可以通过将子类实例转型为某个超类类型来访问到被隐藏的域，在这个超类中该域未被隐藏。如果你想让这个程序打印Derived，也就是说，你想展示覆写行为，那么你可以用公共方法来替代公共域。在任何情况下，这都是一个好主意，因为它提供了更好的封装[EJ Item 19]。下面的程序版本就使用了这项技术，并且能够打印出我们所期望的Derived： class Base { public String getClassName() { return "Base". } } class Derived extends Base { public String getClassName() { return "Derived". } } public class PrivateMatter { public static void main(String[] args) {
```

`System.out.println(new Derived().getClassName()). }` } } 请注意，我们将Derived类中的getClassName方法声明成了public的，尽管在最初的程序中与其相对应的域是私有的。就像前面提到的那样，覆写方法的访问修饰符与它要覆写的方法的访问修饰符相比，所具有的限制性不能有任何降低。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