

Java更多的类谜题69：黑色的渐隐 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_138617.htm 假设你不能修改前一个谜题（谜题68）中的X和C这两个类。你能否编写一个类，其main方法将读取X.Y类中的Z域的值，然后打印它。注意，不能使用反射。本谜题初看起来是不可能实现的。毕竟，X.Y类被具有相同名字的一个域给遮掩了，因此对其命名的尝试将引用到该域上。事实上，我们是可以引用到一个被遮掩的类型名的，其技巧就是在某一种特殊的语法上下文环境中使用该名字，在该语法上下文环境中允许出现一个类型但是不允许出现一个变量。在转型表达式的括号中间的部分就是这样一种上下文环境。下面的程序通过使用这种技术解决了这个谜题，并且将打印出我们所期望的Black：

```
public class FadeToBlack {
public static void main(String[] args){
System.out.println(((X.Y)null).Z). } }
```

请注意，我们是用一个具有X.Y类型的表达式来访问X.Y类的Z域的。就像我们在谜题48和54中所看到的，用一个表达式而不是类型名来访问一个静态成员是合法的，但却是一种有问题的用法。不借助这种有问题的用法，而是通过在一个类声明的extends子句中使用一个被遮掩的类这种方式，你也可以解决本谜题。因为基类总是一种类型，出现在extends子句中的名字从来都不会被解析为变量名。下面的程序就展示了这项技术，它也会打印出Black：

```
public class FadeToBlack { static class Xy extends X.Y{ }
public static void main(String[ ] args){ System.out.println(Xy.Z). } }
```

如果你使用的5.0或更新的版本，那么通过在一个类型变量声

明的extends子句中使用X.Y这种方式，你也可以解决本谜题：

```
public class FadeToBlack { public static void main(String[] args){  
System.out.println(T.Z). } }
```

总之，要解决由类型被变量遮掩而引发的问题，需要按照标准的命名习惯来重命名类型和变量，就像在谜题68中所讨论的那样。如果做不到这一点，那么你应该在只允许类型名的上下文环境中使用被遮掩的类型名。幸运的话，你将永远不需要凭借这种对程序的变形来解决问题，因为大多数的类库作者都很明智，他们都避免了必需使用这种变形的有问题的用法。然而，如果你确实发现自己身处这种境地，那么你最好是要了解这个问题需要解决。

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