

JAVA更多的类谜题71：进口税 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_138611.htm 在5.0版中，Java平台引入了大量的可以使操作数组变得更加容易的工具。下面这个谜题使用了变量参数、自动包装、静态导入（请查

看<http://java.sun.com/j2se/5.0/docs/guide/language> [Java-5.0]）以及便捷方法Arrays.toString（请查看谜题60）。那么，这个程序会打印什么呢？

```
import static java.util.Arrays.toString; class
ImportDuty { public static void main(String[] args) { printArgs(1, 2,
3, 4, 5). } static void printArgs(Object... args) {
```

```
System.out.println(toString(args)). } }
```

你可能会期望该程序打印[1,2,3,4,5]，实际上它确实会这么做，只要它能编译。令人沮丧的是，看起来编译器找不到恰当的toString方法：

```
ImportDuty.java:9:Object.toString()can ' t be applied to(Object[])
System.out.println(toString(args)). ^ 是不是编译器的理解力太差了？为什么它会尝试着去应用Object.toString()呢？它与调用参数列表并不匹配，而Arrays.toString(Object[])却可以完全匹配。编译器在选择在运行期将被调用的方法时，所作的第一件事就是在肯定能找到该方法的范围内挑选[JLS 15.12.1]。编译器将在包含了具有恰当名字的方法的最小闭合范围内进行挑选，在我们的程序中，这个范围就是ImportDuty类，它包含了从Object继承而来的toString方法。在这个范围中没有任何可以应用于toString(args)调用的方法，因此编译器必须拒绝该程序。换句话说，我们想要的toString方法没有在调用点所处的范围内。导入的toString方法被ImportDuty从Object那里继
```

承而来的具有相同名字的方法所遮蔽（shade）了[JLS 6.3.1]。遮蔽与遮掩（谜题68）非常相像，二者的关键区别是一个声明只能遮蔽类型相同的另一个声明：一个类型声明可以遮蔽另一个类型声明，一个变量声明可以遮蔽另一个变量声明，一个方法声明可以遮蔽另一个方法声明。与其形成对照的是，变量声明可以遮掩类型和包声明，而类型声明也可以遮掩包声明。当一个声明遮蔽了另一个声明时，简单名将引用到遮蔽声明中的实体。在本例中，toString引用的是从Object继承而来的toString方法。简单地说，本身就属于某个范围的成员在该范围内与静态导入相比具有优先权。这导致的后果之一就是与Object的方法具有相同名字的静态方法不能通过静态导入工具而得到使用。既然你不能对Arrays.toString使用静态导入，那么你就应该用一个普通的导入声明来代替。下面就是Arrays.toString应该被正确使用的方式：

```
import java.util.Arrays; class ImportDuty { static void printArgs(Object... args) { System.out.println(Arrays.toString(args)). } }
```

如果你特别强烈地想避免显式地限定Arrays.toString调用，那么你可以编写你自己的私有静态转发方法：

```
private static String toString(Object[] a) { return Arrays.toString(a). }
```

静态导入工具所专门针对的情况是：程序中会重复地使用另一个类的静态元素，而每一次用到的时候都进行限定又会使程序变得乱成一锅粥。在这类情况中，静态导入工具可以显著地提高可读性。这比通过实现接口来继承其常量要安全得多，而实现接口这种做法是你从来都不应该采用的 [EJ Item 17]。然而，滥用静态导入工具也会损害可读性，因为这会使得静态成员的类在何处被使用显得非常不清晰。应该有节制地使用静态导入

，只有在非常需要的情况下才应该使用它们。对API设计者来说，要意识到当某个方法的名字已经出现在某个作用域内时，静态导入工具并不能被有效地作用于该方法上。这意味着静态导入不能用于那些与通用接口中的方法共享方法名的静态方法，而且也从来不能用于那些与Object中的方法共享方法名的静态方法。再次说明一下，本谜题所要说明的仍然是你在覆写之外的情况中使用名字重用通常都会产生混乱。我们通过重载、隐藏和遮掩看清楚了一点，现在我们又通过遮蔽看到了同样的问题。 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