

JAVA异常谜题37:极端不可思议 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\\_E5\\_B\\_C\\_82\\_E5\\_B8\\_B8\\_c97\\_138517.htm](https://www.100test.com/kao_ti2020/138/2021_2022_JAVA_E5_B_C_82_E5_B8_B8_c97_138517.htm) 本谜题测试的是你对某些规则的掌握程度，这些规则用于声明从方法中抛出并被catch语句块所捕获的异常。下面的三个程序每一个都会打印些什么？不要假设它们都可以通过编译：

```
import java.io.IOException.  
public class Arcane1 { public static void main(String[] args) { try {  
System.out.println("Hello world"). } catch(IOException e) {  
System.out.println("I ' ve never seen println fail!"). } } } public class  
Arcane2 { public static void main(String[] args) { try { // If you have  
nothing nice to say, say nothing } catch(Exception e) {  
System.out.println("This can ' t happen"). } } } interface Type1 {  
void f() throws CloneNotSupportedException. } interface Type2 {  
void f() throws InterruptedException. } interface Type3 extends  
Type1, Type2 { } public class Arcane3 implements Type3 { public  
void f() { System.out.println("Hello world"). } public static void  
main(String[] args) { Type3 t3 = new Arcane3(). t3.f(). } } 第一个程  
序，Arcane1，展示了被检查异常的一个基本原则。它看起来  
应该是可以编译的：try子句执行I/O，并且catch子句捕  
获IOException异常。但是这个程序不能编译，因为println方  
法没有声明会抛出任何被检查异常，而IOException却正是一  
个被检查异常。语言规范中描述道：如果一个catch子句要捕  
获一个类型为E的被检查异常，而其相对应的try子句不能抛  
出E的某种子类型的异常，那么这就是一个编译期错误[JLS  
11.2.3]。基于同样的理由，第二个程序，Arcane2，看起来应
```

该是不可以编译的，但是它却可以。它之所以可以编译，是因为它唯一的catch子句检查了Exception。尽管JLS在这一点上十分含混不清，但是捕获Exception或Throwable的catch子句是合法的，不管与其相对应的try子句的内容为何。尽管Arcane2是一个合法的程序，但是catch子句的内容永远的不会被执行，这个程序什么都不会打印。第三个程序，Arcane3，看起来它也不能编译。方法f在Type1接口中声明要抛出被检查异常CloneNotSupportedException，并且在Type2接口中声明要抛出被检查异常InterruptedException。Type3接口继承了Type1和Type2，因此，看起来在静态类型为Type3的对象上调用方法f时，有潜在可能会抛出这些异常。一个方法必须要么捕获其方法体可以抛出的所有被检查异常，要么声明它将抛出这些异常。Arcane3的main方法在静态类型为Type3的对象上调用了方法f，但它对CloneNotSupportedException和InterruptedException并没有作这些处理。那么，为什么这个程序可以编译呢？上述分析的缺陷在于对“Type3.f可以抛出在Type1.f上声明的异常和在Type2.f上声明的异常”所做的假设。这并不正确，因为每一个接口都限制了方法f可以抛出的被检查异常集合。一个方法可以抛出的被检查异常集合是它所适用的所有类型声明要抛出的被检查异常集合的交集，而不是合集。因此，静态类型为Type3的对象上的f方法根本就不能抛出任何被检查异常。因此，Arcane3可以毫无错误地通过编译，并且打印Hello world。总之，第一个程序说明了一项基本要求，即对于捕获被检查异常的catch子句，只有在相应的try子句可以抛出这些异常时才被允许。第二个程序说明了这项要求不会应用到的冷僻案例。第三个程序说明了多

个继承而来的throws子句的交集，将减少而不是增加方法允许抛出的异常数量。本谜题所说明的行为一般不会引发难以捉摸的bug，但是你第一次看到它们时，可能会有点吃惊。

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](http://www.100test.com)