

JAVA异常谜题40:不情愿的构造器 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_BC_82_E5_B8_B8_c97_138511.htm 尽管在一个方法声明中看到

一个throws子句是很常见的，但是在构造器的声明中看到

一个throws子句就很少见了。下面的程序就有这样的一个声明。那么，它将打印出什么呢？

```
public class Reluctant { private Reluctant internalInstance = new Reluctant(). public Reluctant() throws Exception { throw new Exception("I ' m not coming out"). }
```

```
public static void main(String[] args) { try { Reluctant b = new Reluctant(). System.out.println("Surprise!"). } catch (Exception ex) { System.out.println("I told you so"). } } }
```

main方法调用

了Reluctant构造器，它将抛出一个异常。你可能期望catch子句能够捕获这个异常，并且打印I told you so。凑近仔细看看这个程序就会发现，Reluctant实例还包含第二个内部实例，它的构造器也会抛出一个异常。无论抛出哪一个异常，看起来main中的catch子句都应该捕获它，因此预测该程序将打印I told you应该是一个安全的赌注。但是当你尝试着去运行它时，就会发现它压根没有去做这类的事情：它抛出

了StackOverflowError异常，为什么呢？与大多数抛出StackOverflowError异常的程序一样，本程序也包含了一个无限递归。当你调用一个构造器时，实例变量的初始化操作将先于构造器的程序体而运行[JLS 12.5]。在本谜题中，internalInstance变量的初始化操作递归调用了构造器，而该构造器通过再次调用Reluctant构造器而初始化该变量自己的internalInstance域，如此无限递归下去。这些递归调用在构

造器程序体获得执行机会之前就会抛出StackOverflowError异常，因为StackOverflowError是Error的子类型而不是Exception的子类型，所以catch子句无法捕获它。对于一个对象包含与它自己类型相同的实例的情况，并不少见。例如，链接列表节点、树节点和图节点都属于这种情况。你必须非常小心地初始化这样的包含实例，以避免StackOverflowError异常。至于本谜题名义上的题目：声明将抛出异常的构造器，你需要注意，构造器必须声明其实例初始化操作会抛出的所有被检查异常。下面这个展示了常见的“服务提供商”模式的程序，将不能编译，因为它违反了这条规则：

```
public class Car {  
    private static Class engineClass = .... private Engine engine =  
    (Engine)enginClass.newInstance(). public Car(){ } } 尽管其构造器  
没有任何程序体，但是它将抛出两个被检查异常  
， InstantiationException和IllegalAccessException。它们是  
Class.newInstance抛出的，该方法是在初始化engine域的时候被  
调用的。订正该程序的最好方式是创建一个私有的、静态的  
助手方法，它负责计算域的初始值，并恰当地处理异常。在  
本案中，我们假设选择engineClass所引用的Class对象，保证它  
是可访问的并且是可实例化的。下面的Car版本将可以毫无  
错误地通过编译： //Fixed - instance initializers don ' t throw  
checked exceptions public class Car { private static Class engineClass  
= .... private Engine engine = newEngine. private static Engine  
newEngine() { try { return (Engine)engineClass.newInstance(). }  
catch (IllegalAccessException e) { throw new AssertionError(e). }  
catch (InstantiationException e) {
```

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