

Java类谜题49：比生命更大 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\\_E7\\_B1\\_BB\\_E8\\_B0\\_9C\\_c97\\_138488.htm](https://www.100test.com/kao_ti2020/138/2021_2022_Java_E7_B1_BB_E8_B0_9C_c97_138488.htm) 假如小报是可信的，那么摇滚之王“猫王”就会直到今天仍然在世。下面的程序用来估算猫王当前的腰带尺寸，方法是根据在公开演出中所观察到的他的体态发展趋势来进行投射。该程序中使用

了Calendar.getInstance().get(Calendar.YEAR)这个惯用法，它返回当前的日历年份。那么，该程序会打印出什么呢？

```
public class Elvis {
    public static final Elvis INSTANCE = new Elvis();
    private final int beltSize;
    private static final int CURRENT_YEAR =
        Calendar.getInstance().get(Calendar.YEAR);
    private Elvis() {
        beltSize = CURRENT_YEAR - 1930;
    }
    public int beltSize() {
        return beltSize;
    }
    public static void main(String[] args) {
        System.out.println("Elvis wears a size " + INSTANCE.beltSize() +
            " belt.");
    }
}
```

第一眼看去，这个程序是在计算当前的年份减去1930的值。如果它是正确的，那么在2006年，该程序将打印出Elvis wears a size 76 belt。如果你尝试着去运行该程序，你就会了解到小报是错误的，这证明你不能相信在报纸到读到的任何东西。该程序将打印出Elvis wears a size -1930 belt。也许猫王已经在反物质的宇宙中定居了。该程序所遇到的问题是

由类初始化顺序中的循环而引起的[JLS 12.4]。让我们来看看其细节。Elvis类的初始化是由虚拟机对其main方法的调用而触发的。首先，其静态域被设置为缺省值[JLS 4.12.5]，其中INSTANCE域被设置为null，CURRENT\_YEAR被设置为0。接下来，静态域初始器按照其出现的顺序执行。第一个静态

域是INSTANCE，它的值是通过调用Elvis()构造器而计算出来的。这个构造器会用一个涉及静态域CURRENT\_YEAR的表达式来初始化beltSize。通常，读取一个静态域是会引起一个类被初始化的事件之一，但是我们已经是在初始化Elvis类了。递归的初始化尝试会直接被忽略掉[JLS 12.4.2, 第3步]。因此，CURRENT\_YEAR的值仍旧是其缺省值0。这就是为什么Elvis的腰带尺寸变成了-1930的原因。最后，从构造器返回以完成Elvis类的初始化，假设我们是在2006年运行该程序，那么我们就将静态域CURRENT\_YEAR初始化成了2006。遗憾的是，这个域现在所具有的正确值对于向Elvis.INSTANCE.beltSize的计算施加影响来说已经太晚了，beltSize的值已经是-1930了。这正是后续所有对Elvis.INSTANCE.beltSize()的调用将返回的值。该程序表明，在final类型的静态域被初始化之前，存在着读取它的值的可能，而此时该静态域包含的还只是其所属类型的缺省值。这是与直觉相违背的，因为我们通常会将final类型的域看作是常量。final类型的域只有在其初始化表达式是常量表达式时才是常量[JLS 15.28]。由类初始化中的循环所引发的问题是难以诊断的，但是一旦被诊断到，通常是很容易订正的。要想订正一个类初始化循环，需要重新对静态域的初始器进行排序，使得每一个初始器都出现在任何依赖于它的初始器之前。在这个程序中，CURRENT\_YEAR的声明属于在INSTANCE声明之前的情况，因为Elvis实例的创建需要CURRENT\_YEAR被初始化。一旦CURRENT\_YEAR的声明被移走，Elvis就真的比生命更大了。某些通用的设计模式本质上就是初始化循环的，特别是本谜题所展示的单例模式

( Singleton ) [Gamma95]和服务提供者框架 ( Service Provider Framework ) [EJ Item 1]。类型安全的枚举模式 ( Typesafe Enum pattern ) [EJ Item 21]也会引起类初始化的循环。5.0版本添加了对这种使用枚举类型的模式的语言级支持。为了减少问题发生的可能性，对枚举类型的静态初始器做了一些限制[JLS 16.5, 8.9]。总之，要当心类初始化循环。最简单的循环只涉及到一个单一的类，但是它们也可能涉及多个类。类初始化循环也并非总是坏事，但是它们可能会导致在静态域被初始化之前就调用构造器。静态域，甚至是final类型的静态域，可能会在它们被初始化之前，被读走其缺省值。

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)