论全世界所有程序员都会犯的错误 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/252/2021_2022__E8_AE_BA_E5_85_A8_E4_B8_96_E7_c104_252519.htm 当年，某国际巨星的“ 龙种 ”曝光，众人指责他对不起娇妻，逼得他出面召开记者会，向世人自白他犯了“ 全世界所有男人都会犯的错误 ”。从来没犯过这种错误的我，也因此常常认为自己不是个男人。 虽然没犯过“ 全世界所有男人都会犯的错误 ”，但是我倒是曾经犯了“ 全世界所有程序员都会犯的错误 ”。不管使用何种语言，全世界所有程序员都一定犯过这种错误，那就是：太依赖编译器，却不知道编译器做了哪些事。 一般来说，越高阶的程序语言，会提供越多语法上的便利，以方便程序撰写，这就俗称为syntactic sugar，我称其为“ 语法上的甜头 ”。虽说是甜头，但是如果你未能了解该语法的实质内涵，很可能会未尝甜头，却吃尽苦头。 不久前，我收到一个电子邮件，读者列出下面的Java程序，向我求救。看过这个程序之后，我确定这又是一个“ 全世界所有程序员都会犯的错误 ”。 程序1 class Singleton { private static Singleton obj = new Singleton(). public static int counter1. public static int counter2 = 0. private Singleton() { counter1 . counter2 . } public static Singleton getInstance() { return obj. } } 程序2 public class MyMain { public static void main(String[] args) { Singleton obj = Singleton.getInstance(). System.out.println("obj.counter1==" obj.counter1). System.out.println("obj.counter2==" obj.counter2). } } 执行结果是： obj.counter1==1 obj.counter2==0 你有没有被此结果吓一跳？乍看程序代码，你很可能会认为counter1

和counter2的值一定 会相等，但执行结果显然不是如此。其实，程序1被编译后的程序应该等同于下面的程序3 ： class Singleton { private static Singleton obj. public static int counter1. public static int counter2. static { // 这就是class constructor // 在进入此class constructor之前，class已经被JVM // 配置好内存，所有的static field都会被先设定为0， // 所以此时counter1和counter2都已经是0， 且singleton为null obj = new Singleton(). // 问题皆由此行程序产生 // counter1不会在此被设定为0 counter2 = 0. // counter2再被设定一次0（其实是多此一举） } private Singleton() { // 这是instance constructor counter1 . counter2 . } public static Singleton getInstance() { return obj. } } 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