JAVA字符谜题13:不劳而获 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_AD_97_E7_AC_A6_c97_138548.htm 下面的程序将打印一个单词，其第一个字母是由一个随机数生成器来选择的。请描述该程序的行为： import java.util.Random. public class Rhymes { private static Random rnd = new Random(). public static void main(String[] args) { StringBuffer word = null. switch(rnd.nextInt(2)) { case 1: word = new StringBuffer(’ P ’). case 2: word = new StringBuffer(’ G ’). default: word = new StringBuffer(’ M ’). } word.append(’ a ’). word.append(’ i ’). word.append(’ n ’). System.out.println(word). } } 乍一看，这个程序可能会在一次又一次的运行中，以相等的概率打印出Pain，Gain或 Main。看起来该程序会根据随机数生成器所选取的值来选择单词的第一个字母：0选M，1选P，2选G。谜题的题目也许已经给你提供了线索，它实际上既不会打印Pain，也不会打印Gain。也许更令人吃惊的是，它也不会打印Main，并且它的行为不会在一次又一次的运行中发生变化，它总是在打印ain。 有三个bug凑到一起引发了这种行为。你完全没有发现它们吗？第一个bug是所选取的随机数使得switch语句只能到达其三种情况中的两种。Random.nextInt(int)的规范描述道：“ 返回一个伪随机的、均等地分布在从0（包括）到指定的数值（不包括）之间的一个int数值 ”[Java-API]。这意味着表达式rnd.nextInt(2)可能的取值只有0和1，Switch语句将永远也到不了case 2分支，这表示程序将永远不会打印Gain。nextInt的参数应该是3而不是2。

这是一个相当常见的问题源，被熟知为“栅栏柱错误（fencepost error）”。这个名字来源于对下面这个问题最常见的但却是错误的答案，如果你要建造一个100英尺长的栅栏，其栅栏柱间隔为10英尺，那么你需要多少根栅栏柱呢？11根或9根都是正确答案，这取决于是否要在栅栏的两端树立栅栏柱，但是10根却是错误的。要当心栅栏柱错误，每当你在处理长度、范围或模数的时候，都要仔细确定其端点是否应该被包括在内，并且要确保你的代码的行为要与其相对应。第二个bug是在不同的情况（case）中没有任何break语句。不论switch表达式为何值，该程序都将执行其相对应的case以及所有后续的case[JLS 14.11]。因此，尽管每一个case都对变量word赋了一个值，但是总是最后一个赋值胜出，覆盖了前面的赋值。最后一个赋值将总是最后一种情况（default），即new StringBuffer{’M’}。这表明该程序将总是打印Main，而从来不打印Pain或Gain。 在switch的各种情况中缺少break语句是非常常见的错误。从5.0版本起，javac提供了-Xlint:fallthrough标志，当你忘记在一个case与下一个case之间添加break语句是，它可以生成警告信息。不要从一个非空的case向下进入了另一个case。这是一种拙劣的风格，因为它并不常用，因此会误导读者。十次中有九次它都会包含错误。如果Java不是模仿C建模的，那么它倒是有可能不需要break。对语言设计者的教训是：应该考虑提供一个结构化的switch语句。 最后一个，也是最微妙的一个bug是表达式new StringBuffer(’M’)可能没有做哪些你希望它做的事情。你可能对StringBuffer(char)构造器并不熟悉，这很容易解释：它压根就不存在。StringBuffer有一个无参数的构造器，一个接受

一个String作为字符串缓冲区初始内容的构造器，以及一个接受一个int作为缓冲区初始容量的构造器。在本例中，编译器会选择接受int的构造器，通过拓宽原始类型转换把字符数值’M’转换为一个int数值77[JLS 5.1.2]。换句话说，new StringBuffer(’M’)返回的是一个具有初始容量77的空的字符串缓冲区。该程序余下的部分将字符a、i和n添加到了这个空字符串缓冲区中，并打印出该字符串缓冲区那总是ain的内容。 为了避免这类问题，不管在什么时候，都要尽可能使用熟悉的惯用法和API。如果你必须使用不熟悉的API，那么请仔细阅读其文档。在本例中，程序应该使用常用的接受一个String的StringBuffer构造器。 下面是该程序订正了这三个bug之后的正确版本，它将以均等的概率打印Pain、Gain和Main：import java.util.Random. public class Rhymes1 { private static Random rnd = new Random(). 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