

Java高级谜题86：有毒的括号垃圾 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_E9_AB_98_E7_BA_A7_c97_138704.htm 你能否举出这样一个合法的Java表达式，只要对它的某个子表达式加上括号就可以使其成为不合法的表达式，而添加的括号只是为了注解未加括号时赋值的顺序？插入一对用来注解现有赋值顺序的括号对程序的合法性似乎是应该没有任何影响的。事实上，绝大多数情况下确实是没有影响的。但是，在两种情况下，插入一对看上去没有影响的括号可能会令合法的Java程序变得不合法。这种奇怪的情况是由于数值的二进制补码的不对称性引起的，就像在谜题33和谜题64中所讨论的那样。你可能会联想到，最小的int型负数其绝对值比最大的int型正数大1

：Integer.MIN_VALUE是-2³¹，即-2,147,483,648，而Integer.MAX_VALUE是2³¹-1，即2,147,483,647。Java不支持负的十进制字面常量；int和long类型的负数常量都是由正数十进制字面常量前加一元负操作符（-）构成。这种构成方式是由一条特殊的语言规则所决定的：在int类型的十进制字面常量中，最大的是2147483648。而从0到2147483647的所有十进制字面常量都可以在任何能够使用int类型字面常量的地方出现，但是字面常量2147483648只能作为一元负操作符的操作数来使用[JLS 3.10.1]。一旦你知道了这个规则，这个谜题就很容易了。符号-2147483648构成了一个合法的Java表达式，它由一元负操作符加上一个int型字面常量 2147483648组成。通过添加一对括号来注解（很不重要的）赋值顺序，即写成-(2147483648)，就会破坏这条规则。信不信由你，下面这个

程序肯定会出现一个编译期错误，如果去掉了括号，那么错误也就没有了：`public class PoisonParen { int i = -(2147483648). }`

类似地，上述情况也适用于long型字面常量。下面这个程序也会产生一个编译期错误，并且如果你去掉括号错误也会消失：`public class PoisonParen { long j = -(9223372036854774808L). }`

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