



面向对象设计与构造第一次作业



第一部分：训练目标



通过对表达式结构进行建模，完成多变量多项式的括号展开，初步体会层次化设计的思想。

第二部分：预备知识

- 1、Java 基础语法与基本容器的使用。
- 2、扩展 BNF 描述的形式化表述。
- 3、正则表达式、递归下降或其他解析方法。

第三部分：题目描述

本次作业需要完成的任务为：读入一个包含加、减、乘、乘方以及括号（其中括号的深度**至多为1层**）的**多变量**表达式，输出**恒等变形展开所有括号后**的表达式。

在本次作业中，**展开所有括号**的定义是：对原输入表达式 E 做**恒等变形**，得到新表达式 E' ，且 E' 中不含有字符（和）。

第四部分：基本概念

一、基本概念的声明

- **带符号整数** 支持前导 0 的十进制带符号整数（若为正数，则正号可以省略），无进制标识。如： $+02$ 、 -16 、 20220928 等。
- **因子**
 - **变量因子**
 - **幂函数**
 - **一般形式** 由自变量 x, y, z ，指数符号 $**$ 和指数组成，指数为一个**非负带符号整数**，如： $x ** +2$ ， $y ** 02$ ， $z ** 2$ 。
 - **省略形式** 当指数为 1 的时候，可以省略指数符号 $**$ 和指数，如： x ， y ， z 。
 - **常数因子** 包含一个带符号整数，如： 233 。
 - **表达式因子** 用一对小括号包裹起来的表达式，可以带指数，且指数为一个**非负带符号整数**，例如 $(x**2 + 2*y + z)**2$ 。表达式的定义将在表达式的相关设定中进行详细介绍。
- **项** 由乘法运算符连接若干因子组成，如 $x * 02$ 。此外，**在第一个因子之前，可以带一个正号或者负号**，如 $+ x * 02$ 、 $- +3 * x$ 。注意，**空串不属于合法的项**。





< 第一次作业 - 2023面向对象设计与构造



以带一个正号或者负号，表示第一个项的正负，如： $- -1 + x ** 233$ 、 $+ -2 + x ** 19260817$ 。注意，空串不属于合法的表达式。

- **空白字符** 在本次作业中，空白字符包含且仅包含空格 `<space>` (ascii 值 32) 和水平制表符 `\t` (ascii 值 9)。其他的空白字符，均属于非法字符。

对于空白字符，有以下几点规定：

- 带符号整数内不允许包含空白字符，注意**符号与整数之间**也不允许包含空白字符。
- 指数运算符内不允许包含空白字符，如 `* *` 不合法。
- 因子、项、表达式，在不与前两条条件矛盾的前提下，可以在任意位置包含任意数量的空白字符。

二、设定的形式化表述

- 表达式 \rightarrow 空白项 [加减 空白项] 项 空白项 | 表达式 加减 空白项 项 空白项
- 项 \rightarrow [加减 空白项] 因子 | 项 空白项 '*' 空白项 因子
- 因子 \rightarrow 变量因子 | 常数因子 | 表达式因子
- 变量因子 \rightarrow 幂函数
- 常数因子 \rightarrow 带符号的整数
- 表达式因子 \rightarrow '(' 表达式 ')' [空白项 指数]
- 幂函数 \rightarrow ('x' | 'y' | 'z') [空白项 指数]
- 指数 \rightarrow '**' 空白项 '[' 允许前导零的整数 (注：指数一定不是负数)
- 带符号的整数 \rightarrow [加减] 允许前导零的整数
- 允许前导零的整数 \rightarrow ('0' | '1' | '2' | ... | '9') {'0' | '1' | '2' | ... | '9'}
- 空白项 \rightarrow {空白字符}
- 空白字符 \rightarrow (空格) | `\t`
- 加减 \rightarrow '+' | '-'

其中

- {} 表示允许存在 0 个、1 个或多个。
- [] 表示允许存在 0 个或 1 个。
- () 内的运算拥有更高优先级，类似数学中的括号。
- | 表示在多个之中选择一个。
- 上述表述中使用单引号包裹的串表示字符串字面量，如 '(' 表示字符 (。

式子的具体含义参照其数学含义。

若输入字符串能够由“表达式”推导得出，则输入字符串合法。具体推导方法请参阅“**第一单元形式化表述说明**”文档。

除了满足上述形式化表述之外，我们本次作业的输入数据的**额外限制**请参见**第五部分：输入/输出说明**的数据限制部分。

第五部分：输入/输出说明

一、公测说明



个人中心



所有课程



我的图床



课程团队



关于



注销

输出格式

输出展开括号之后的表达式。

数据限制

- 输入表达式**一定满足**基本概念部分给出的**形式化描述**。
- 输入表达式中**至多包含1层括号**。
- 对于规则“指数 \rightarrow ****** 空白项 带符号的整数”，我们保证**此处的带符号整数中不会出现 - 号**，且保证**输入数据的指数最大不超过 8**。
- 在表达式化简过程中，如果遇到了需要计算 $0^{**}0$ 的值进行化简的这种情况，默认 $0^{**}0 = 1$ 。
- 输入表达式的有效长度至多为 200 个字符。其中输入表达式的有效长度指的是输入表达式去除掉所有空白符后剩余的字符总数。

判定模式

本次作业中，对于每个测试点的判定分为**正确性判定**和**性能判定**。其中，正确性判定为 80 分，性能判定部分为 20 分，二者之和为总分。

注意：**获得性能分的前提是，在正确性判定环节被判定为正确**。如果被判定为错误，则性能分部分为0分。

正确性判定：

- 输出的表达式须符合表达式的形式化描述、**不能包含括号**且与原表达式**恒等**。
- 本次作业中对于恒等的定义：设 $f(x)$ 的定义域为 D_1 ， D_1 包含于 R ， $g(x)$ 的定义域为 D_2 ， D_2 包含于 R ，对任意 $x \in D_1 \cap D_2$ ， $f(x) = g(x)$ 成立。

性能判定：

- 在本次作业中，性能分的唯一评判依据是**输出结果的有效长度**，有效长度的定义在**数据限制部分**已经给出。
- 设某同学给出的**正确答案**的有效长度为 L_p ，目前**所有人**给出的正确答案中有效长度**最小的**为 L_{min} 。

记 $x = \frac{L_p}{L_{min}}$ ，则该同学**性能分百分比**为：

$$r(x) = 100\% \cdot \begin{cases} 1 & x = 1 \\ -31.8239x^4 + 155.9038x^3 - 279.2180x^2 + 214.0743x - 57.9370 & 1 < x \leq 1.5 \\ 0 & x > 1.5 \end{cases}$$

举例来说，就是这样：

x	$r(x)$
1.0	100.0%
1.05	79.9%
1.1	60.5%
1.2	29.0%
1.3	10.9%
1.4	4.5%
1.5	0.0%



二、互测说明



互测时，你可以通过提交**输入数据**和**期望得到的正确的输出**，该组数据会被用来测试同一个互测房间中的其他同学的程序。输入数据必须符合上述的文法规则,并且满足代价函数要求。提交的输出只需要包含一行，即输出的正确表达式。



数据限制



- 输入表达式**一定满足**基本概念部分给出的**形式化描述**。
- 输入表达式中**至多包含1层括号**。
- 对于规则“指数 \rightarrow ****** 空白项 带符号的整数”，保证**此处的带符号整数中不会出现 - 号**，且保证**输入数据的指数最大不超过 8**。
- 输入表达式的有效长度至多为 50 个字符。其中输入表达式的有效长度指的是输入表达式去除掉所有空白符后剩余的字符总数。（本条与公测部分的限制不同）
- 除此之外，为了限制不合理的 hack，我们要求输入表达式的代价 $\text{Cost}(\text{Expr}) \leq 10000$ ，其中表达式代价的计算方法如下（本条与公测部分的限制不同）：



代价函数

- $\text{Cost}(\text{常数}) = \max(1, \text{len}(\text{常数}))$ （常数的前导零不算在其长度内）
- $\text{Cost}(x) = \text{Cost}(y) = \text{Cost}(z) = 1$
- $\text{Cost}(a + b) = \text{Cost}(a - b) = \text{Cost}(a) + \text{Cost}(b)$
- $\text{Cost}(a * b) = \text{Cost}(a) * \text{Cost}(b)$ （多项相乘时从左到右计算）
- $\text{Cost}(a ** b) =$
 - 若a是单变量因子， $\text{Cost}(a ** b) = 1$
 - 若a是表达式因子(c)， $\text{Cost}(a ** b) = \max(\text{Cost}(c), 2) ^ \max(b, 1)$
- $\text{Cost}(+a) = \text{Cost}(-a) = \text{Cost}(a) + 1$


如果提交的数据不满足上述数据限制，则该数据将被系统拒绝，且不会用来对同屋其他被测程序进行测试。

三、样例

#	输入	输出(省略标签)	说明
1	1	1	根据表达式定义可得。
2	x	x	根据表达式定义可得。
3	$x+2*y$	$x+2*y$	未合并同类项，但表达式依然等价。
4	$x+2*x+z+y$	$3*x+y+z$	在表达式等价的基础上合并同类项。
5	$-3*(x)$	$-3*x$	拆括号后得到符合输出格式的表达式。
6	$-3*(y-1)$	$-3*y--3*1$	拆括号后得到符合输出格式的表达式。
7	$(z-1)**2$	$z**2-2*z+1$	拆括号后得到符合输出格式的表达式。
8	$(x+2*y+1)**0$	1	0 次幂的值为 1。
9	$(x+1)**3+1$	$x**3+3*x**2+3*x+2$	拆括号后得到符合输出格式的表达式。



 < 第一次作业 - 2023面向对象设计与构造 			
12	$x*(x-(x+1))$	Wrong Format! (无需输出)	嵌套括号不会出现在本次作业中，本次作业不需考虑。

 注意：由于本作业可被判定为正确的答案不唯一，以上样例的输出**仅保证正确性，但并不一定为性能最优解**。



第六部分：设计建议

- 在 Java 内，不建议使用静态数组。推荐使用 `ArrayList`、`HashMap`、`HashSet` 一类的数据结构，快速管理和调配手中无序的数据。
- 对于使用一般输入手动解析的同学，处理字符串时可以考虑使用**正则表达式**，相关的 API 可以了解 `Pattern` 和 `Matcher` 类；另外也可以考虑使用**递归下降**等方法进行解析，相关教程我们会放在实验或者训练中。
- 对于每种表达式结构，可以考虑单独建一个类，把整个表达式构建为树形结构。
- 本次作业为单层括号，但建议**提前考虑**多层括号将如何架构与迭代。

第七部分：提示与警示

一、提示

- Java 内的原生整数类型有 `long` 和 `int`，长度分别为 64 位和 32 位，遇到整数过大的问题，可以使用 `BigInteger` 存储。
- 不要重复造轮子！不要重复造轮子！不要重复造轮子！重要的事情说三遍**
- 我们鼓励大家通过 Baidu、Google、Stack Overflow 等方式自行学习和解决问题。
- 如果还有更多的问题，请到讨论区提问。但是**请善用讨论区**，并在此之前认真阅读包括但不限于课程要求文档、指导书、搜索引擎结果等的**内容**。[关于如何提问](#)。

二、警示

- 如果在互测中发现其他人的代码疑似存在**抄袭**等行为，可向课程组举报，课程组感谢同学们为 OO 课程建设所作出的贡献。