



面向对象设计与构造第三次作业



第一部分：训练目标



通过对表达式结构进行建模，完成多项式的括号展开与函数调用、化简，进一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

本次作业中需要完成任务为：读入一系列自定义函数的定义以及一个包含幂函数、三角函数、自定义函数调用、求导算子的表达式，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' 。其中， E' 中不再含有自定义函数，不再含有求导算子，且只包含必要的括号（必要括号的定义见公测说明-正确性判定）。

第四部分:迭代内容概览

在第一次作业基础上，本次迭代作业增加了以下几点：

- 本次作业支持求导操作,新增求导算子。
 - 根据第三部分形式化表述，求导因子可以出现在很多位置，包括函数调用实参，函数定义表达式，三角函数内部等，注意考虑周全。
 - 为了限制难度，在输入中，求导算子最多只能出现一次，具体见第六部分-数据限制。
- 本次作业函数表达式中支持调用其他“已定义的”函数（保证不会出现递归调用，具体见第六部分-数据限制）。
- 本周实验会着重指导求导将如何层次化实现。

第五部分：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、19260817 等。
- 因子
 - 变量因子
 - 幂函数



- **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`。
 - **三角函数**
 - **一般形式** 类似于幂函数, 由 `sin(<因子>)` 或 `cos(<因子>)`、指数符号 `**` 和指数组成, 其中:
 - 指数为符号不是 `-` 的整数, 如: `sin(x) ** +2`。
 - **省略形式** 当指数为 1 的时候, 可以采用省略形式, 省略指数符号 `**` 和指数部分, 如: `sin(x)`。
 - 本指导书范围内的“三角函数”**仅包含 sin 和 cos**。
 - **自定义函数**
 - 自定义函数的**定义**形如 `f(x, y, z) = 函数表达式`, 比如 `f(y) = y**2`, `g(x, y) = sin(x)*cos(y)`, `h(x, y, z) = x + y + z`。
 - `f`、`g`、`h` 是函数的**函数名**。在本次作业中, 保证函数名**只使用 f, g, h**, 且**不出现同名函数的重复定义** (因此每次最多只有 3 个自定义函数)。更具体的约束信息请看第六部分中的数据限制部分。
 - `x`、`y`、`z` 为函数的**形参**。在本次作业中, **形参个数为 1~3 个**。形参**只使用 x, y, z**, 且同一函数定义中不会出现重复使用的形参。
 - 函数表达式为一个关于形参的表达式。函数表达式的一般形式参见**形式化定义**。
 - 自定义函数的**调用**形如 `f(因子, 因子, 因子)`, 比如 `f(x**2)`, `g(sin(x), cos(x))`, `h(1, 0, -1)`。
 - 因子 为函数调用时的**实参**, 包含任意一种因子。
 - **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, 且指数为符号不是 `-` 的整数, 例如 `(x**2 + 2*x)**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, 可以带一个正号或负号**, 如 `+ x * 02`、`- +3 * x`。注意, **空串不属于合法的项**。
 - **表达式** 由加法和减法运算符连接若干项组成, 如:
`(-1 + x ** 233)* sin(x**2) ** 06 - cos(x) * 3 * sin(x)`。此外, **在第一项之前, 可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 ** 1113`。
 - **求导因子(新增)**, 由 `dx(表达式)`, `dy(表达式)`, `dz(表达式)` 三种算符构成, 分别代表对表达式的`x`变量求导, 对表达式的`y`变量求导, 对表达式的`z`变量求导。由于本单元表达式自变量有三个分别为`x,y,z`, 因此**这里的求导可以理解为偏导**。例如对`x`求导时, 对表达式中的其他变量(`y`与`z`)视作常量对待。例如
 - `x+dx(x+sin((2*y+cos(z))))` 展开后等于 `x+1`
- 此外, 为了保证**结果一致性**, 自定义函数内包含求导因子, 当调用该函数时, 先将自定义函数表达式求导后再代入实参。例如函数 `h(x) = dx(x)`, 自定义函数调用 `h(sin(x)) = 1` 而不是 `h(sin(x))=cos(x)`。
- **空白字符** 在本次作业中, 空白字符仅包含空格 `<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, 注意带符号整数本身的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。
 - 函数保留字内不允许包含空白字符, 即 `sin`, `cos`, `dx`, `dy`, `dz` 关键字内不可以含有空白字符。

二、设定的形式化表述



第三次作业 - 2023面向对象设计与构造



- 变量因子 → 幂函数 | 三角函数 | 自定义函数调用
- 常数因子 → 带符号的整数



- 表达式因子 \rightarrow '(' 表达式 ')' [空白项 指数]
- 幂函数 \rightarrow 自变量 [空白项 指数]



- 自变量 \rightarrow 'x' | 'y' | 'z'



- 三角函数 \rightarrow 'sin' 空白项 '(' 空白项 因子 空白项 ')' [空白项 指数] | 'cos' 空白项 '(' 空白项 因子 空白项 ')' [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 `[+|-]` 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 `'+'|'-'`

自定义函数相关(相关限制见“公测数据限制”)

- 自定义函数**定义** → 自定义函数名 空白项 '(' 空白项 自变量 空白项 '[' 空白项 自变量 空白项 '[' 空白项 自变量 空白项 ']' 空白项 ')' 空白项 '=' 空白项 函数表达式
- 自定义函数**调用** → 自定义函数名 空白项 '(' 空白项 因子 空白项 '[' 空白项 因子 空白项 '[' 空白项 因子 空白项 ']' 空白项 ')'
- 自定义函数名 → 'f' | 'g' | 'h'
- 函数表达式 → 表达式 (注: 本次作业函数表达式中可以调用其他自定义函数, 但保证不会出现递归调用的情况)

求导算子相关(相关限制见“公测数据限制”)

- 求导因子 \rightarrow 求导算子 空白项 '(' 空白项 表达式 空白项 ')'
- 求导算子 \rightarrow 'dx' | 'dy' | 'dz'

形式化表述中 $\{ \} [()]$ 符号的含义已在第一次作业指导书中说明, 不再赘述。

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**第六部分：输入/输出说明 的数据限制部分**。

三、求导公式

本次作业可能用到的求导公式有：

I. 当 $f(x) = c$ (c 为常数) 时, $f'(x) = 0$

II. 当 $f(x) = x^n$ ($n \neq 0$) 时, $f'(x) = nx^{n-1}$

III. 当 $f(x) = \sin(x)$ 时, $f'(x) = \cos(x)$

IV. 当 $f(x) = \cos(x)$ 时, $f'(x) = -\sin(x)$



个人中心



所有课程



我的图床



课程团队



关于



● 注销

第六部分：输入/输出说明

一、公测说明

输入格式

本次作业的输入数据包含若干行：

- 第一行为一个整数 n ($0 \leq n \leq 3$)，表示自定义函数定义的个数，最多3个。
- 第 2 到第 $n + 1$ 行，每行为一行字符串，表示一个自定义函数的定义。
- 第 $n + 2$ 行，一行字符串，表示待展开表达式。

输出格式

输出展开括号之后，不再含有自定义函数，不再含有求导算子，且只包含必要的括号的表达式。（必要括号的定义见公测说明-正确性判定）。

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。
- 为了限制难度，输入数据中，求导因子最多只能出现一次。
- 自定义函数定义满足以下限制：
 - 不会出现重名函数。
 - 函数表达式与上次作业不同，允许调用其他已定义的自定义函数（保证不会出现递归调用的情况），下面是几个不合法的例子。
 - 函数定义时 $f(x,y) = f(x,2)+y+1$,出现递归调用，不合法。
 - 函数定义时 $f(x,y) = g(x,y)+y, g(x,y)=f(x,y)+y$, f 先定义, g 后定义, f 在定义时调用了未定义的函数 g ，不合法。
 - 函数形参不能重复出现，即无需考虑 $f(x,x)=x**2+x$ 这类情况
 - 函数定义式中出现的变量都必须在形参中有定义
- 对于规则“指数 \rightarrow $**$ 空白项 $[+]$ 允许前导零的整数”，我们本次要求输入数据的指数不能超过 8。
- 在表达式化简过程中，如果遇到了 $0**0$ 这种情况，默认 $0**0 = 1$ 。
- 为了避免待展开表达式或函数表达式过长。最后一行输入的待展开表达式的有效长度至多为 200 个字符，每个单个自定义函数定义的有效长度至多为 150 个字符。其中有效长度指的是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，需要展开所有括号且与保持原表达式恒等。

与 $\cos((x*x))$)。(注意是“不带指数”的表达式因子, 如果是 $\sin((x+1)**2)$,这**并不符合必要括号**的定义, 你必须将其展开为 $\sin((x**2+2*x+1))$ 这种类似的形式才是合法的)

- 例如 $\sin(1)$ 与 $\sin((1))$ 均为展开形式, 但 $\sin(((1)))$ 不是, 因为后者除了函数调用和三角嵌套表达式因子的一层括号外, 还包括了表达式内嵌套表达式的括号
- 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%

该答案得到的性能分即为 $r(x) \times 20$ 。

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。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**。
- 为了限制难度, 输入数据中, 求导因子最多只能出现一次。
- 自定义函数限制与公测数据限制相同。
- 对于规则“指数 \rightarrow ****** 空白项 **[+]** 允许前导零的整数”, 我们本次要求**输入数据的指数不能超过 8**。
- 最终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)
- 自定义函数定义的有效长度至多30个字符, 有效长度定义同上



第三次作业 - 2023面向对象设计与构造



Cost(常数) = max(1, len(常数)) (常数的长度等于其类型长度)

- $\text{Cost}(x) = \text{Cost}(y) = \text{Cost}(z) = 1$
- $\text{Cost}(a + b) = \text{Cost}(a - b) = \text{Cost}(a) + \text{Cost}(b)$
- $\text{Cost}(a * b) = \text{Cost}(a) * \text{Cost}(b)$ (多项相乘时从左到右计算)
- $\text{Cost}(\sin(a)) = \text{Cost}(\cos(a)) = \text{Cost}(a) + 1$
- $\text{Cost}(a ** b) =$
 - 若a是单变量因子, $\text{Cost}(a ** b) = 1$
 - 若a是表达式因子(c), $\text{Cost}(a ** b) = \max(\text{Cost}(c), 2) ^ \max(b, 1)$
 - 若a是三角函数因子, $\text{Cost}(a ** b) = 2 ^ b + \text{Cost}(a)$
- $\text{Cost}(+a) = \text{Cost}(-a) = \text{Cost}(a) + 1$
- $\text{Cost}(h) = \text{Cost}(h') * 2$, h是自定义函数调用, 其中 h' 是将调用 h 的参数作为表达式因子代入后, 所得到的表达式, 同时注意h的实参代价不能超过阈值500。
- $\text{Cost}(dx(a)) = \text{Cost}(dy(a)) = \text{Cost}(dz(a)) = 2 ^ \text{Cost}(a)$

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	说明
1	1 f(x)=x**2 (x+f(x))*x+dy(cos(sin(y)))	x**2+x**3-cos(y)*sin(sin(y))	带入 f(x)之后, 求导可得结果
2	0 dy((x+1*(x+2*(x+3))))	0	对y求导, 将x看作常数, 常数求导为0
3	1 f(z)=z*(sin(z)) dx(sin(((x+1)**2-1))+cos(y))	2*x*cos((x**2+2*x))+2*cos((x**2+2*x))	三角函数内部利用链式法则求导
4	1 f(x,y)=dx((x+(x+1-x))**2+y) f(sin(x),y)+cos(z)	2*sin(x)+2+cos(z)	函数定义也可以带求导因子,注意先将自定义函数表达式求导后, 再代入实参
5	1 f(x,y,z)=x+y+z f(x,y,(dx(sin(x))+1))	x+y+cos(x)+1	函数调用实参包含求导因子
6	1 f(x)=x*(sin(x)+cos(x)) dx(x-dy(f(y)+x))	Wrong Format	求导算子只能出现一次, 本作业不需考虑

第七部分：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。



个人中心



所有课程



我的图床



课程团队



关于



注销



- 注意对某一变量求导时，将其他两个变量看作常数处理。



第八部分：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。

