

# 面向对象第一单元第三次作业指导书

## 摘要

本次作业，需要完成的任务为**包含简单幂函数和简单正余弦函数的导函数及其组合**的求解。

## 问题

## 设定

首先是一些基本概念的声明：

- **带符号整数** 支持**前导0**的十进制带符号整数，符号可省略，无进制标识，如：`+02`、`-16`、`19260817` 等。

- **因子**

- **变量因子**

- **幂函数**

- **一般形式** 由自变量 $x$ ，指数符号 `**` 和指数组成，指数为一个带符号整数，如：`x**+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`。

- 允许**嵌套因子**的三角函数，

本次作业支持**因子嵌套在三角函数里面，并整体作为一个因子**，即一个因子作为另一个三角函数因子的自变量。(该因子包括上下文提到的所有因子，包括三角函数自身)

有以下两种形式

- **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：

- `<因子>` 指任意一个因子，例如 `x**2`，`x`，`233`，`(x**2 + 2*x)`。

- 指数为一个带符号整数，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)`。

此种因子的完整例子有：`sin(x**2)**3`，`cos(sin(x))`，`sin(x)`，`sin(sin(cos(cos(x**2))))**2` 以及 `cos((x**2 + 2*x))` 等。

- 本指导书范围内**所有的词语“三角函数”**，除非特殊说明，否则**一律包含且仅包含 `sin` 和 `cos` 两个函数**

- **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，例如 `(x**2 + 2*x)`，表达式的定义将在表达式的相关设定中进行详细介绍。

- **提示：**

- **表达式因子不支持幂运算**，例如 `(1113*x)**62621` **不是合法的因子**。
      - **三角函数因子不允许出现指数为变量的情况**，指数依然只能是带符号整数，例如 `sin(x)**sin(x)` **是不合法的**，因为**指数不能是自变量**。

- **项**

- **一般形式**由乘法运算符连接若干任意因子组成，如：`x * cos(x) * x`，`sin(x ** 2) * cos(sin(x)) ** 2 * x ** 2` 等。
  - **因子**的个数至少为一个
  - 提示：**项内因子不仅仅是同类因子**
- **特殊形式**
  - 第一个因子为常数因子，值为1 且**总共不止有一个因子**的时候，可以省略该常数因子或表示为正号开头的形式，如：`x ** 2 * x ** 1`、`+ x ** 2`、`+ cos(x) * cos(x)`、`sin(x) * cos(x)`。
  - 第一个因子为常数因子，值为-1 且**总共不止有一个因子**的时候，可以表示为负号开头的形式，如：`-x ** 2`、`- cos(x) * sin(x)`。
- **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(sin(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`。此处有几点注意：
  - **项**的个数至少为一个
  - 关于**表达式因子**：表达式可以作为因子，其定义为被一对小括号包裹起来的表达式，即我们允许诸如 `(x * cos(x))` 这种式子的出现
    - 提示：表达式因子可以递归嵌套，例如 `sin( ( x**2 + sin((1 + 3*x)) ) )`
- **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。
- 三角函数的保留字内不允许包含空白字符，即 `sin`，`cos` 关键字内不可以含有空白字符。
- 因子、项、表达式中，在不与上两条矛盾的前提下，可以在任意位置包含任意数量的空白字符。

## 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 | 三角函数
- 常数因子  $\rightarrow$  带符号的整数
- 表达式因子  $\rightarrow$  '(' 表达式 ')'
- 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]
- 幂函数  $\rightarrow$  `x` [空白项 指数]
- 指数  $\rightarrow$  `**` 空白项 带符号的整数
- 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}
- 空白字符  $\rightarrow$  (空格) | `\t` (水平制表符)
- 空白项  $\rightarrow$  {空白字符}
- 加减  $\rightarrow$  + | -

其中{}表示0个、1个或多个，[]表示0个或1个，|表示多个之中选择。

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，则为正确输入字符串，否则为 WRONG FORMAT。具体推导方法请参阅“第一单元形式化表述说明”文档。

## 描述

求导是数学计算中的一个计算方法，它的定义就是，当自变量的增量趋于零时，因变量的增量与自变量的增量之商的极限。

在本次作业中，要判断输入的表达式是否合法（即符合上述的表达式基本规则），并进行求导计算，进行化简并输出它的导函数。

本次作业可能用到的求导公式有：

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

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

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

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

V . 当  $f(x) = \arctan(x)$  时,  $f'(x) = 1/(1+x^2)$  (曾经有, 本次作业取消)

VI . 链式法则:  $[f(g(x))]' = f'(g(x))g'(x)$

VII . 乘法: 当  $f(x) = g(x)h(x)$  时,  $f'(x) = g'(x)h(x) + g(x)h'(x)$

## 一些规定

- 一个表达式可能有多个解释。因此，对于一个表达式，只要存在一条合法解释，该表达式即为合法，且我们保证以这些解释在数学意义上均相等。
- 为了方便评测机评测，我们限制表达式的指数（包括幂函数和三角函数）的**绝对值不超过 50**（十进制），超过此范围需要输出 `WRONG FORMAT!`
- 我们将 `0**0` 定义为 `1`（例如 `sin(0)**0`）（即c标准库和python中的定义）。

## 判定

### 输入格式

输入中，包含且仅包含一行，表示一个表达式

### 输出格式

- 如果是一组合法的输入数据，则应当输出一行，表示求出的导函数。格式同样需要满足上述的表达式基本格式规则，其中要求带符号的整数均为**十进制形式**。
- 如果是一组不合法的输入数据，则应当输出一行 `WRONG FORMAT!`

### 数据限制

- 输入数据的最大长度为50。**（请注意，这里不是有效长度，是去除右侧换行符后的总长度）。

上述限制被定义为**数据基本限制**。在测试中，不会出现不符合该限制的数据。在此范围限制内，强测不作其他任何限制。

## 判定模式

### 正确性判定

对于这次作业结果正确性的判定，在输出符合格式要求的前提下，我们采用如下的方式：

- 在区间 $[-10, 10]$ 上，进行1000次线性随机选取，设得到的数为 $\{x_i\}$  ( $1 \leq i \leq 1000$ )
- 设输入多项式为 $f(x)$ ，其导函数为 $f'(x)$ （即正确答案，由sympy进行符号运算），将 $\{x_i\}$ 依次代入 $f'(x)$ ，得到结果 $\{a_i\}$
- 设待测输出的多项式为 $g'(x)$ ，将 $\{x_i\}$ 依次代入 $g'(x)$ ，得到结果 $\{b_i\}$
- 将数列 $\{a_i\}$ 和数列 $\{b_i\}$ 依次进行比较，判定每个数是否依次相等
- 如果全部相等，则认为该组输出正确，否则认为错误

其中

- 在比较两个数的时候，判定是否相等的依据是：对于数 $a_i$ 表示正确结果， $b_i$ 表示根据输出计算出的结果，若满足

$$\frac{|a_i - b_i|}{\max(|a_i|, 1)} < 10^{-8}$$

则视为 $a_i$ 与 $b_i$ 相等。

- 考虑到可能会出现随机出的数位于无意义点上导致计算出错，故在上述计算 $b_i$ 的过程中，如果第 $i$ 个数计算错误，则将重新生成该 $x_i$ ，并重新计算。每个数最多将会重试五次，如果重试次数达到上限后依然无法正常计算，则判定该组输出错误。举例说明的话，就是表达式 $\frac{x^2}{x}$ 最终也会被判定为和表达式 $x$ 等价。（由 $a_i$ 不可计算引发的重新计算将不被计算在这五次内）

综上，简而言之，你可以理解为：只要是和标准结果等价的表达式（允许定义域上的点差异），都会被认定为正确答案。

## 性能分判定

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。

有效长度定义为，输出结果去除所有的空白字符（<space>、\t）后的长度，设为 $L$ 。

设某同学给出的正确答案的有效长度为 $L_p$ ，所有人目前给出的正确答案的有效长度的最小值为 $L_{min}$ ，最大值为 $L_{max}$ ，平均值为 $L_{avg}$ 。

设

$$base_{min} = p \cdot L_{avg} + (1 - p) \cdot L_{min},$$

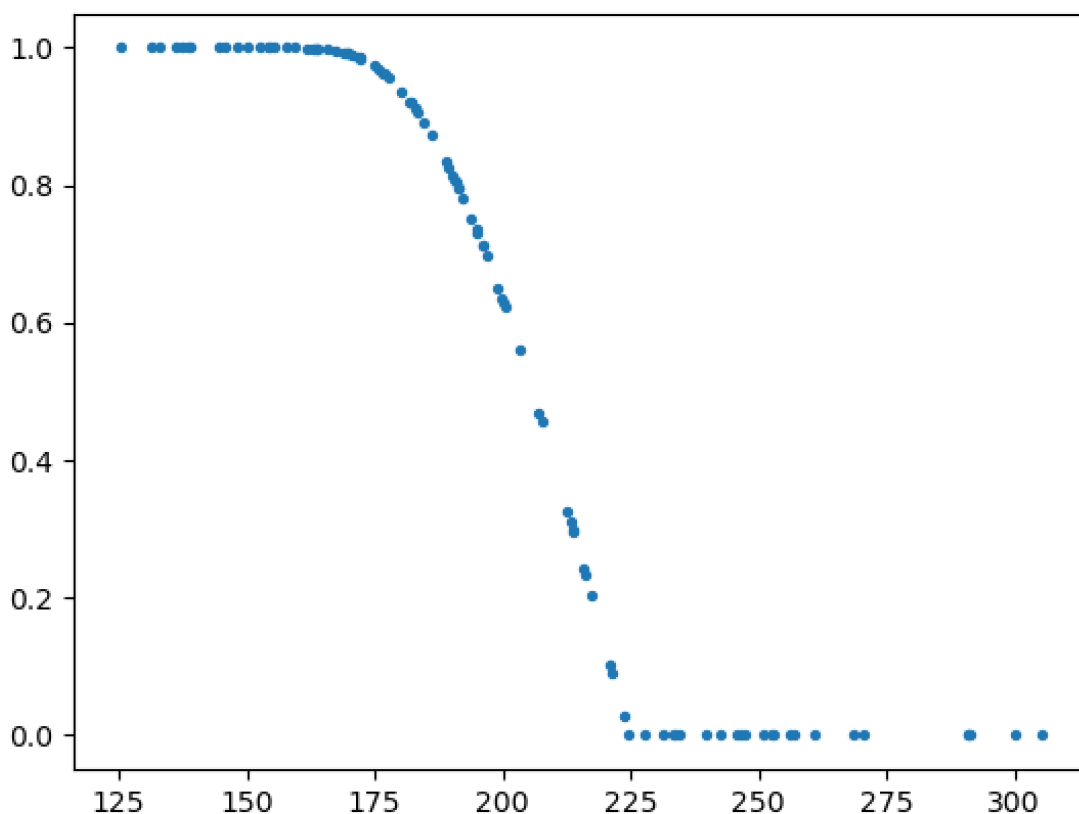
$$base_{max} = p \cdot L_{max} + (1 - p) \cdot L_{avg},$$

$$p = 0.25$$

则该同学性能分百分比为：

$$r(L_p) = 100\% \cdot \begin{cases} 1 & L_p \leq base_{min} \\ 1 - 10^{1 - \frac{base_{max} - base_{min}}{L_p - base_{min}}} & base_{min} < L_p \leq base_{max} \\ 0 & L_p > base_{max} \end{cases}$$

以下是用正态分布生成的一组数据，模拟的性能分得分情况，



在格式错误的情况下，输出是固定的，所以对于格式错误的数据点，只要被判定为正确即可获得100%的性能分。

需要注意的是，**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。

## 互测相关

在**互测**环节

- 数据须符合**数据基本限制**
- 数据需要保证**表达式一定合法**。（即，互测环节不允许输出为 `WRONG FORMAT!` 的数据）
- 对于符合表达式格式的输入数据，其正确导函数在 $[-10, 10]$ 的范围内不存在值超过 $10^{20}$ 的点。
- 所有指数一定严格大于0的。

对于限制的第三条，互测时判定方式如下：

- 在区间 $[-10, 10]$ 上，线性随机选取5000个数，设为 $\{x_i\}$  ( $1 \leq i \leq 5000$ )
- 设输入多项式为 $f(x)$ ，其导函数为 $f'(x)$ （即正确答案，由 `sympy` 进行符号计算并求导），将 $\{x_i\}$ 依次代入 $f'(x)$ ，得到结果 $\{a_i\}$ 。（使用 `sympy`，使用合适的精度进行计算）
- 如果满足 $\forall 1 \leq i \leq 5000, |a_i| \leq 10^{20}$ ，则输入数据判定为满足这一要求，否则判定为不满足。

如果不满足上述数据基本限制，或者未通过上述方式的评判的，该数据将被系统忽略，不会对被测程序进行测试。

## 样例

#	输入	输出	解释
1	1	0	根据表达式定义可得。
2	$4x+x^{**2}+x+1$	$2*x+5$	幂函数与常量因子为同类因子
3	$4x+x^{**2}+x$	$4+2*x+1$	未合并同类项，但表达式依然等价。
4	$4x+x^{**2}+x$	WRONG FORMAT!	$4x$ 不是合法项，应该写作 $4*x$
5	$-4*x + x^{**2} + x$	$2*x+5$	$-4*x$ 为合法项，且表达式第一项前也可以包含正负号。
6	$+4*x - x^{**2} + x$	$2*x+5$	$-x^{**2}$ 为合法项。
7	$+19260817*x$	19260817	根据表达式定义可得。
8	$+ 19260817*x$	19260817	多项式第一项前可以带有正负号。
9	$+++19260817*x$	WRONG FORMAT!	项内有符号整数不可以包含空白字符。
10	$+++1$	0	合法，实际上三个加号分别对应的是表达式、项、带符号整数。
11	$+++x$	WRONG FORMAT!	不合法，用现有文法无法正确解释。
12	$++++1$	WRONG FORMAT!	因子里头只能省略第一个，且必须在包含多个因子时才可以省略。
13	$+++++++$	WRONG FORMAT!	根据表达式定义可得。
14	$1926\ 0817\ *x$	WRONG FORMAT!	项内有符号整数不可以包含空白字符。
15	(空)	WRONG FORMAT!	空串不属于合法表达式。
16	$2*\sin(x)$	$2*\cos(x)$	常数因子与三角函数相乘
18	$-1*\cos(x)$	$\sin(x)$	同上
18	$23+\sin(x)*3$	$3*\cos(x)$	同上
19	$\cos(x)*\sin(x)*5+4*x^{**3}$	$-5*\sin(x)^{**2} + 5*\cos(x)*\cos(x)+12*x^{**2}$	根据表达式定义可得。
20	$43+4*x^{**3}$	$12*x^{**2}$	仅包括常数因子项加包括幂函数的同类因子项
21	$6*\sin(x)$	WRONG FORMAT!	'sin'间不能有空白字符
22	$6*\cos(x)$	WRONG FORMAT!	'cos'间不能有空白字符
23	$6*\arctan(x)$	WRONG FORMAT!	无法识别的函数名(arctan已经取消了)

#	输入	输出	解释
24	$1 \cdot x^2$	$2 \cdot x$	根据表达式定义可得。
25	$\sin(\cos(x))$	$-\cos(\cos(x)) \cdot \sin(x)$	三角函数作为三角函数因子
26	$\sin((2 \cdot x)) \cdot (\cos(x)+1)$	$2 \cdot (\cos(x) + 1) \cdot \cos((2 \cdot x)) - \sin(x) \cdot \sin((2 \cdot x))$	$(2 \cdot x)$ 是表达式因子。
27	$\sin((2 \cdot x)) \cdot (\cos(x)+1)^2$	WRONG FORMAT!	表达式因子不得带有幂运算
28	$\sin(x)$	$\cos(x)$	根据表达式定义可得。
29	$\sin(x)$	$\cos(x)$	根据表达式定义可得。
30	$\sin(x)^{50}$	$50 \cdot \cos(x) \cdot \sin(x)^{49}$	根据表达式定义可得。
31	$\sin(x)^{51}$	WRONG FORMAT!	指数绝对值一概不得超过50（十进制）
32	$\sin(2 \cdot x)$	WRONG FORMAT!	' $2 \cdot x$ '不是因子
33	$1113^{50}$	WRONG FORMAT!	幂函数的自变量只能为x

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

## 补充信息

### 关于评测

- 评测时，会自动忽略掉行末的空格以及文件末多余的回车。
- 对于输入，如果包含多行，则忽略第一行以后的内容即可。（由于不忽略第一行以后的内容导致的错误，一概后果自负）
- 类似的，对于输出结果，如果包含多行，则在评判的时候将忽略第一行以后的内容。（也就是说，你们可以在正文之后附加一些其他的信息以改善自己调试的体验。以及，由于不在第一行输出正确答案造成的错误，也一概后果自负）

### 一点点的提示

- Java内的原生整数类型有 `long` 和 `int`，长度分别为64位和32位。
- 如果觉得上述数据类型不够用的话，可以百度一下Java内可以怎样快速处理这个问题。
- 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。
- 关于输入字符串的处理，推荐使用**正则表达式**，相关的API可以了解 `Matcher` 和 `Pattern` 类。
- 这次作业，看上去似乎很难，其实找对了方法后并不难。关键思想是，**化整为零**，可以这样考虑
  - 对于每一种函数（常数、幂函数、三角函数），建立类
  - 对于每一种函数组合规则（乘法、加减法、嵌套），建立类
  - 对于上述的两种类，均实现一个求导接口
    - 其中，第一种类，做法显而易见
    - 其中，第二种类，做法一样显而易见
  - 通过上述两种类及其求导接口，把整个表达式构建为树结构，进行链式求导。

- 对于秒掉正确性部分后，想要最大限度优化性能的夫佬同学，一样可以将上述的化整为零思想作为可行思路之一，设计算法。

## 一点点想说的话

- **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、Stackoverflow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、搜索引擎结果等的內容。
- 如果想要深入了解Java的一些内置数据结构的特性和原理，推荐以下方法：
  - 查阅官方文档
  - 阅读该部分源代码（另外，在Idea内，Ctrl+左键点击方法名、变量名、类名、包名，有惊喜）
- **请大家把重点放在架构的设计上，以及如何正确完成功能上。性能部分，学有余力再去仔细研究，做一些力所能及的优化。**