# Readme

### 一、 功能说明

该程序能实现任意多项的一元多项式加减运算,并能编译执行,且能根据输入得到相应正确(或预期)的结果。

## 二、 运行环境及指令规范说明

运行环境: JDK 1.8 Eclipse

指令规范: Eclipse 内运行程序后直接在 console 内进行输入(一行), 回车即可。

### 三、 输入说明

#### 标准输入格式:

输入仅有一行,由"*0-9 + - , () {}*"这几种字符和"*空格*"、"制 表符"组成,输入者通过键入回车键确认输入结束。标准的输入由若干多项式、多项式之前的正负号或两个多项式之间的一个正负号,以及若干空格、制表符组成,相邻两个多项式之间有且仅有一个'+'或'-',第一个多项式之前可能存在正负号(若有,则仅有一个),每个多项式内的数对中的 n 都 不相同。数字之前可以有正负号(若有,则仅有一个)。

一个多项式的标准格式为:  $\{(c_1, n_1), (c_2, n_2), \cdots, (c_m, n_m)\}$ ,最外端由花括号与其它多项式区分开,内部是若干由括号包含的数对,括号之间由一个逗号','分隔,两个数之间也由一个','分隔。

在本次编程过程中,对于数对(c,n),c 为系数(输入系数允许为 0),有 $-10^6$ <c< $10^6$ , n 为幂,有 0<=n< $10^6$ , 即 c 最大为 6 位十进制整数,n 最大为 6 位十进制整数。所有数均可能出现前导 0,但 c 和 n 除去符号位之外的长度分别不超过 6 和 6。

输入时,每个多项式限制为最多50个数对,多项式的个数限制为最多20个。

输入实例:  $\{(3,0),(2,2),(12,3)\}+\{(3,1),(-5,3)\}-\{(-199,2),(29,3),(10,7)\}$ 

#### 输入错误时响应信息:

若不合乎输入规范,出现括号不匹配或其他字符或超出限制的情况,响应信息均为:warning!:输入不合法。(小括号内逗号前后即指数和系数只能为直接数字,不能是运算式)

若出现同一多项式内指数相同的情况,响应信息为:warning!:出现同一多项式中有相同指数的情况。

若输入汉字,响应信息为: warning!: 请不要输入汉字。

若输入为空或仅有空格和制表符,响应信息为:warning!:输入不能为空。

## 四、 输出说明

#### 输出规范:

程序的正常运行结果为一个多项式表达式。标准格式为 $\{(c_1,n_1),(c_2,n_2),\cdots,(c_m,n_m)\}$ 。每一个小括号表示一个单项式,多项式内单项式按照单项式次数进行升序排序,且系数为 0 的单项式也会打印出来。

# 五、 程序控制流程图

入口 main 函数



从控制台读入表达式并去除表达 式字符串中的空格和制表符。判 断字符串是否为空



调用 isEnglish 方法判断字符串中 是否含有汉字,若没有进行下一步;否则,则输出警告信息并结束 程序



调用 check 方法判断表达式在输入格式下是否合法, 若合法进行下一步;否则,则输出警告信息并结束程序



匹配表达式中的大括号,即每个 多项式,将每个多项式放入多项 式数组 pol



循环匹配每个多项式中的小括号,即单项式,将数字提取出来分别放入系数和指数数组,同时进行运算



得到结果数组,选择排序将两个 数组按照指数进行升序排序



输出结果, 结束程序