

# JAVA 说明文档

## 一、程序功能说明

该程序功能为一元多项式加减运算，能根据输入的多项式算式计算出正确结果，并输出计算之后的多项式。

## 二、程序运行环境

JAVA 版本: 1.8.0\_111

Eclipse 版本: Mars.1 Release (4.5.1)

## 三、输入说明

### 1.标准输入格式

控制台输入，输入仅有一行，由“0-9+-,(){}”这几种字符和“空格”组成，输入者通过键入回车键确认输入结束。以上字符均为英文样式。

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；若没有，则视为正）。每个多项式内的数对中的  $n$  都不相同（即使系数为 0）。

一个多项式的标准格式为： $\{(c_1, n_1), (c_2, n_2), \dots, (c_m, n_m)\}$ ，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。

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

### 2.输入限制

对于数对  $(c, n)$ ， $c$  为系数，有  $-106 < c < 106$ ， $n$  为幂，有  $0 \leq n < 106$ ，即  $c$  最大为 6 位十进制整数， $n$  最大为 6 位十进制整数。

为了表示负数，允许  $c$  使用“-”作为“负号”，但“-”不可当做“减号”使用（会报“unexpected”）； $n$  使用“-”会提示“非法负号错误”或“unexpected”（因此“-0”也为非法输入）。 $c$  和  $n$  都不允许使用“+”（会报“unexpected”）。故  $c$  和  $n$  都只能是纯数字，不可为算式。

所有数均可以出现前导 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，输入总长度限制为 100000 个字符。

不允许输入空多项式{}，不允许省略  $c$  或  $n$ ，否则当做格式有误处理。

## 四、输出说明

### 1.标准输出

程序的正常运行结果为一个多项式表达式，其规则与输入相同。如果运行结果为空，则输出“{}”。

输出样例： $\{(3, 0), (3, 1), (201, 2), (-22, 3), (10, 7)\}$

### 2.错误提示

#### (1) 错误种类

在实际运行中，若出现多种错误，则只显示其中一种。

“overlong input”:输入字符串过长

“illegal end”:提前中断输入，造成信息不完整（包括直接回车）

“unexpected character 'x'”:输入格式有误，x 为不正确符号

“coeff out of range”:系数 c 超出范围

“degree out of range”:幂 n 超出范围

“illegal negative sign for degree”:幂 n 使用负号

“same degrees”:同一个多项式中出现两个具有相同幂 n 的数对

“too many terms”:数对数量过多

“too many polynomials”:多项式数量过多

## (2) 错误格式

不同种类错误格式不尽相同: 有些错误只会显示错误种类, 有些错误会告知具体的位置。

下面是告知具体位置情况下的显示格式:

“Error: info at x.”:第 x 个非空格字符附近出现了 info 错误 (x 从 1 始计)。

“After having read x polynomials and y extra terms.”:已经读取了 x 个完整的多项式外加 y 个数对 (y 个数对不包含在 x 个多项式中)

## 五、程序控制流程图

