Readme

1. 程序功能说明

任意多项的一元多项式加减运算的程序，并能够编译执行，且能根据输入得到相应正确（或预期）的结果。

有，在给定和时计算的结果。

1. 程序运行所需环境和运行指令规范

本程序在win 10 下JDK 1.8.0\_92，eclipse版本：Neon.2 Release (4.6.2)运行

1. 程序的输入说明
2. 输入格式和限制

本程序不允许使用文件输入。

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，输入的字符串长度理论上无限制，本小节后半部对此有说明。

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），每个多项式内的数对中的n都不相同。

一个多项式的标准格式为：{(c1,n1),(c2,n2),…,(cm,nm)}，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，有-106<c<106, n为幂，有0<=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。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。

此外，每一个多项式之前允许输入+-号，比如下面所列示例：

+{(3,0), (2,2), (12,3)} + {(3,1), (-5,3)} – {(-199,2), (29,3),(10,7)}

-{(3,0), (2,2), (12,3)} + {(3,1), (-5,3)}

{(3,0), (2,2), (12,3)} - {(3,1), (-5,3)}

都是合法的。

1. 各种错误信息及可能的原因
2. "your input is beyond the upper bound of poly's length"

如果输入过长或者单纯是因为多项式过长，就会产生此错误，正常情况下，经过计算，除去空格之后50\*20的多项式最长长度不会超过18100

1. "Your input has illegal characters like …”

此处输出的like之后的字符就是输入中包含的非法字符（不包含在“***0-9 + - , ( ) { }*** ”之中的字符）之一

1. "illegal input found at the begining"

在多项式开头检测到了不合规定的输入，有多种原因会导致此错误，比如只输入字符串“(”，“()”或“-}”等

1. "illegal input found at the end"

经过删空格处理后的输入字符串，结尾只要不符合“)}”的格式，就会报错

1. "the input is not a legal poly"

在读取输入阶段发生的错误会被捕捉至此，包括按下Ctrl+Z

1. "Match failed: your input is not legal"

在经过以上检测步骤之后，任何不能匹配的输入都会产生这个错误

1. "no comma is allowed before the first term of a poly"

类似如下的输入会产生错误

“{,(1,2)}”

1. "Error in term …” ": Exponent should be a positive"

某一项中指数不应为负数

1. "Some unknown error occured during calculate the polynomial"

为了保证程序不会崩溃，在处理多项式计算时，加入了catch，会接到来自于匹配和计算过程中所有的error。

1. 程序计算结果的输出规格

输出规格和输入一致，所有系数为0的项不会输出，按照指数升序排列，如：

输入：{(2,2),(5,6),(-3,4)}+{(-1,2),(2,7),(-2,7)}

输出：{(1,2),(-3,4),(5,6)}

1. 程序控制流程图

