OO第一次作业要求

2017/3/1

# 作业目标

采用命令式语言（c语言）和对象式语言（java语言），针对同一个数学计算公式的实现，分别编制两种不同风格的程序，并通过对比，加深对象式语言的理解。同时逐渐熟悉java语言编程环境，学习java语言，增强java编程能力。

# 作业内容和成果物

## 2.1 作业内容

编写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。

按照上述要求，使用c语言编写实现一个过程式程序。

按照上述要求，使用java语言编写实现一个对象式程序。

## 2.2 提交内容

1）c语言程序（.c文件）；

2）c语言说明文档（Readme，word文件），包含程序控制流图；

3）java语言程序（java程序文件），以及输入内容和相应输出的结果（txt文件，相当于自我测试的案例）；

4）java语言说明文档（Readme，word文件），内容包括：

1. 程序功能说明；
2. 程序运行所需环境和运行指令规范；
3. 程序的输入说明，包括标准输入格式、输入限制和遇见输入错误时的响应信息；
4. 程序计算结果的输出规格，以及可预见的运行错误响应信息；
5. 程序控制流程图。

# 作业要求和限制

## 3.1 程序运行时输入及其标准规格

程序运行所需的输入由控制台或命令行输入，输入方式由程序设计者决定，但是要求在使用文档中加以明确说明。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。

注意，所有空格应该被直接忽略。对于出现其它字符程序应给出相应，对于诸如(3+,1-7)这样的输入程序也应给出响应，并在说明书中予以说明。

输入实例：

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

## 3.2 输出规范

程序的正常运行结果为一个多项式表达式，要求能够把计算结果尽可能清晰地展示给用户，可参考输入时采用的单个多项式表达方式，并要求按照多项式内的单项式次数进行升序排序。

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

相关的输出规范要在**说明文档**中加以说明。

## 3.3 数据结构要求

Java编程必须使用数组，此条为硬性要求，但是未规定数组的使用方法；

使用C语言编程时对使用数据结构无限制。

# 其它说明事项

## 4.1 输入要求

对于更多的细节的输入规范，如与文档冲突，请在readme说明，若没有说明且与文档的冲突，测试者有理由质疑.

## 4.2 输入输出容错与响应

假如输入不满足输入规范，可以自定义处理办法，并在**说明文档**中写明。

这里列举如下几种处理方法，仅供参考：

1. 输出错误提示（明确指出是什么错误更佳）；
2. 输出一个符合输出规范的答案（不要求答案正确，可以自行设计容错处理）；
3. 明确类似{(1,)}为不合法输入。而类似空多项式如何处理应由开发者自行决定；
4. 要考虑输入字符串中出现不合法字符的情况；
5. 要考虑输入字符串中合法字符不合理组合的情况；
6. 其它会引发异常的情况。

## 4.3 说明文档要求

使用word文档必须清除个人信息。即，清除word作者姓名，否则按照故意暴露个人信息行为处理。

## 4.3 环境建议

优先推荐使用Eclipse。

为了方便，请大家提交代码时，按照package名提交。

假设你的工程包名为：package oo.app;

则提交的文件结构为：

--oo

----app

------yourCode.java

------…

即目录结构与包名一致。

# 评判事项

1. 最后测试仅进行Java程序的测试，编写C程序主要是为了让大家感受OO程序和非OO程序在编写时的差异。但也请各位同学对自己负责，认真对待。
2. 任何情况下，程序都**不应crash**，要正常结束（exitcode=0）。

# 其它补充说明

暂无