
OO 第一次作业要求

2018 版

1. 作业目标

采用过程式语言（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。

2. 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。

多项式基本定义如下：

有 $f_i(x) = \sum_{n_i=0}^{\infty} (c_{n_i} x^{n_i})$, $i, n_i, c_{n_i} \in N$ ，在给定 i, n_i, c_{n_i} 和 $x(x \in N)$ 时计算

$f_i(x) \pm f_j(x) \pm \dots$ 的结果。

按照上述要求，使用 c 语言编写实现一个过程式的程序。

按照上述要求，使用 java 语言编写实现一个面向对象式的程序。

2.2 提交内容

- 1) c 语言程序（.c 文件）；
- 2) c 语言说明文档（即 **Readme**，要求为 **PDF 格式文件**），包含程序控制流程图；
- 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 语言程序的说明文档（**Readme**，要求为 **PDF 格式文件**），内容包括：
 - a) 程序功能说明；（即这个程序能够做什么事情）
 - b) 程序运行所需环境和运行指令规范；（即如何运行这个程序）
 - c) 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；（注：即运行程序后如何为程序提供输入信息，以及输入不同的信息后，程序的响应是什么）
 - d) 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；（注：如何理解你的输出内容；在什么情况下，你的程序无法根据给定的输入计算出结果，你的程序表现可能是什么）
 - e) 程序控制流程图。（或是类图）

3. 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，具体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+-,(){}**”这几种半角字符和“**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（即可以处理无限多的多项式），本小节后半部对此有说明。

标准的输入由多组代表多项式的符号集合组成，形式如下“{多项式 1}+{多项式 2}+……+{多项式 n}”。每组花括号内为一个多项式字符串，**所有花括弧前通过‘+’或‘-’代表多项式之间的加减关系**，若第一组花括号前没有则默认为‘+’。

花括号内的多项式由数对(c,n)组成，其中 c 为系数，n 代表次数。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组成，括号之间由一个逗号‘,’分隔，两个数之间也由一个‘,’分隔。一个多项式内的数对中的 n 都不相同。

组成全部多项式的字符串中可以存在空格，但是所有**空格应该被直接忽略**。如 $\{(c_1, n_1), (c_2, n_2)\}$ 这样的表达是被允许的，如果出现‘1 24’则被当成‘124’

在本次编程过程中，对于数对(c,n)，c 为系数，为十进制整数，取值范围为 $-999999 \leq c \leq 999999$ （即 c 前面可以有‘-’）；n 为该项的幂，为十进制整数，有 $0 \leq n \leq 999999$ 。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，输入的字符串在出现非规定的其它字符时，程序应给出报错响应，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。

参考的输入实例：

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

3.2 输出规范

程序的正常运行结果输出为一个**标准的多项式表达式字符串**，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。

如果在输出时有说明性信息（可以理解为注释性的说明），则**该行必须在第一个字符处填入‘#’**，否则将认为该行为某个输出。本规定为适应在线评判的要求，**违反本规定的结果将很有可能导致自动测试时结果被扣分！**

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

任何非负数字之前不得有加号，如+3 或者+0 是被认为错误输出！-0 也是错误输出！**违反本规定的结果将很有可能导致自动测试时结果被扣分！**

对于本程序不能识别的输入内容，必须按照以下规格进行输出，**否则按照错误结果扣分！**

ERROR

#所对应的错误输入内容

相关的输出规范要在**说明文档**（README）中加以说明。

3.3 数据结构要求

Java 编程中，必须使用数组来管理多项式数据，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。

4. 其它说明事项

4.1 输入要求

对于更多的细节的输入规范，应尽量遵守本文档的指导性原则，如与本文档冲突，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) 给出输入错误的提示（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。
- 4) 若结果为空多项式，即所有的项系数为 0，则输出 0；

-
- 5) 要考虑输入字符串中出现不合法字符的情况；
 - 6) 要考虑输入字符串中合法字符不合理组合的情况；
 - 7) 其它会引发异常的情况。

4.3 说明文档要求

必须提交 PDF 文档，不可使用 word 文档，提交 word 文档者视为无效文档。

4.4 环境建议

使用 Eclipse 开发。

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。

4.5 Tip

- 1) 本次作业的一个重要部分是：如何编写简洁高效的输入内容规格检查和内容提取部分的代码。希望大家可以自行学习和参考如何使用“正则表达式”完成相应的功能。
- 2) 不要按照过程式的编程思路，写成从头到尾就一个执行过程的程序。

5. 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）。

6. 其它补充说明

暂无