程序设计实验

多项式计算器

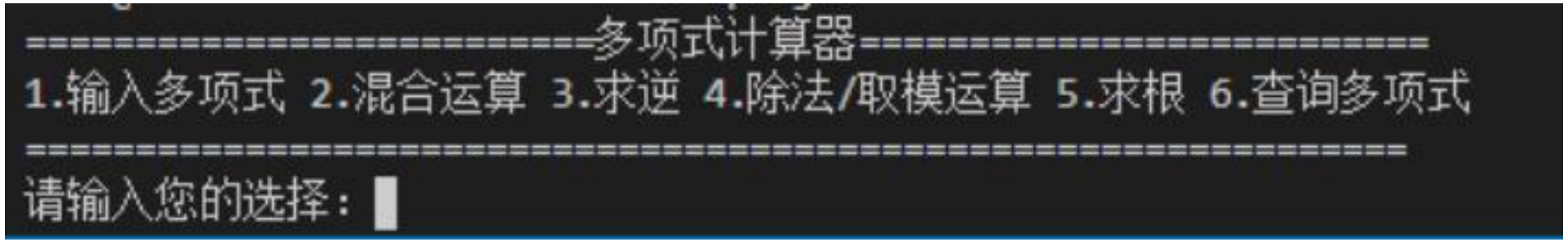
- 专业：计算机科学与技术
- 学号：191220177
- 姓名：周俊杰

目录

01

需求分析

需 求 分 析



在linux环境下做一个多项式计算器

能处理输入的多项式
能对多项式进行混合运算
求逆、取模运算
求根运算
对当前已经有的多项式进行查询

既然是一个计算器，体现良好的交换性能

如果用户输入的多项式是格式错误的，能够给予提示
对于刚拿到这个计算器的人也能够在提示下进行
对用户做的每一步都需要有确认环节

02

数据结构分析

用来存储多项式

设计了一个类，内有数据成员如下：

①长度

②vector<double>存储每一项的系数

数据结构分析

在控制类中

因为需要对用户输入的多项式存储起来并与字符串相对应

用于存放对应关系的数据结构

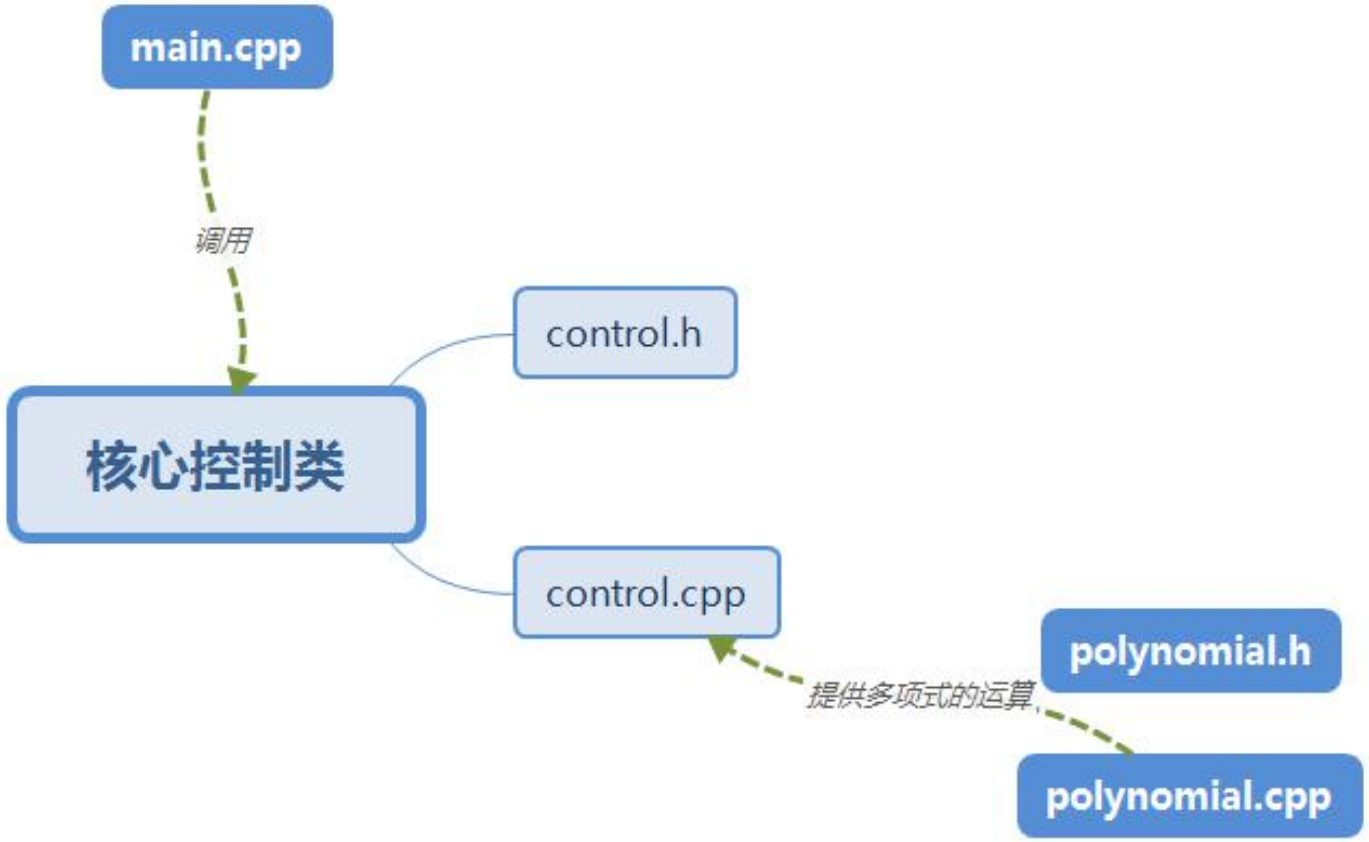
```
map<string,int> ans;
vector<Polynomial> data;
```

map用来存储string 与int之间构成的关系
data则是存放多项式
假设已经知道name,用data[map[name]]获得多项式

03

模块划分

模块划分


main.cpp
调用
control.h
核心控制类
control.cpp
polynomial.h
提供多项式的运算
polynomial.cpp

模块划分

main.cpp

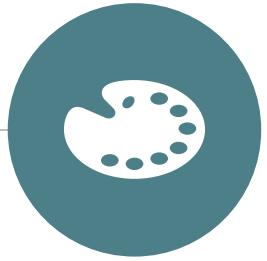
调用control类的实例

包括界面的展出
是输入的字符串与多项式的计算的桥梁

control.cpp

polynomial.cpp

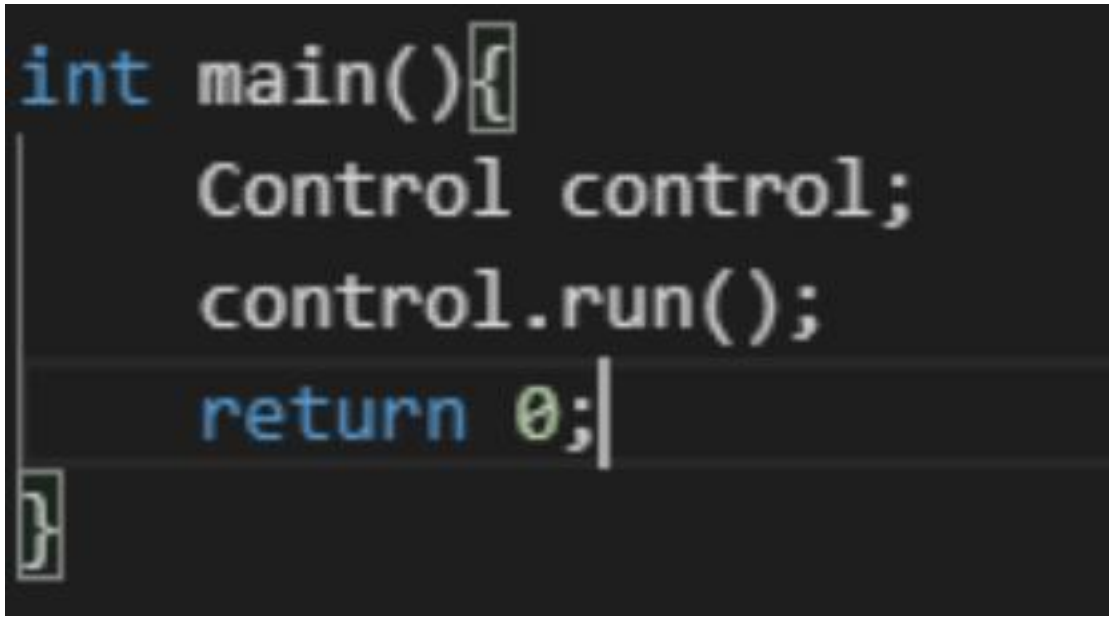
polynomial类的成员函数的实现，包括*+ 求导、积分函数的实现



模块划分

01

main.cpp



模块划分

02

control.cpp

```
class Polynomial
{
public:
    Polynomial();
    Polynomial(const Polynomial& p);//拷贝构造函数
    Polynomial(const double w[], const int n);//传入n及w函数的构造函数
    Polynomial operator+(const Polynomial& p)const;//对+进行重载
    Polynomial operator*(const Polynomial& p);//对*进行重载
    Polynomial derivative() const;//求导
    Polynomial integrate(double a,double b) const;//积分
    Polynomial inversion();//求逆元
    pair<Polynomial,Polynomial> divide(Polynomial& p);//除法
    pair<bool,double> get();//获得根，第一个数字代表是否有根，第二代表根是啥
    double operator()(double x) const;//重载()直接计算f(x)
    friend Polynomial operator*(const double x,const Polynomial& p);//数字和多项式的乘法
    friend ostream& operator <<(ostream &os, Polynomial& p); //重载<<便于输出
private:
    Polynomial integrate() const;
    void fft(int limit,complex *a,int type);
    double a[100];
    int num;
};
```

此处输入标题

```
class Control{
public:
    Control();
    void run();
private:
    map<string,int> ans;
    vector<Polynomial> data;
    void init();
    bool IfRepeatDo(string oper); //询问是否重复做当下的事情 oper为当下要做的事的中文
    int manage(string st);
    void putinto(); //输入多项式
    void calculate(); //对输入的表达式进行计算
    void putout(); //输出多项式
    void divide();//求多项式除法的
    void inversion();//求多项式逆元的
    bool helpexist(string name,string help);//判断这个name存不存在并输出对应help的错误信息
    void getright();//得到多项式的根
    bool judgest(string);//判断是否满足$[a,b]的形式
    pair<double,double> dealst(char *str); //对传入的$[a,b]型分解
    double char_to_num(char* s);//将字符型数据转换成double型数据的函数
};
```

03

polynomial.cpp

04

核心算法/功能

采取的核心算法/功能

对应次方系数相加
注意细节即可

为了提高效率
采用FFT

直接采用定义来求
出逆元

除法可以转化为乘
法

牛顿求根法

多项式加法

多项式乘法

多项式求逆元

多项式除法/取模

多项式求根

多项式计算

采取的核心算法/功能

混合运算

- 混合运算我先采用两个栈一个是操作符栈，一个是数据栈
- 遇到一个符号就判断是操作符还是数据
  - 如果遇到是数据直接压入数据栈
  - 如果遇到操作符，先取出当前栈内的第一个操作符，判断优先级别然后再弹出相应的数据栈内部的数据，进行运算
- 遇到$符号特殊处理一下：扫描直到]结束，然后将整个$[a,b]直接压入操作符栈中
- 直到读到最后一个字符为止
- 最后再对栈扫描一遍，直到操作符栈为空
- 在扫描的同时判断字符串内部是否合法

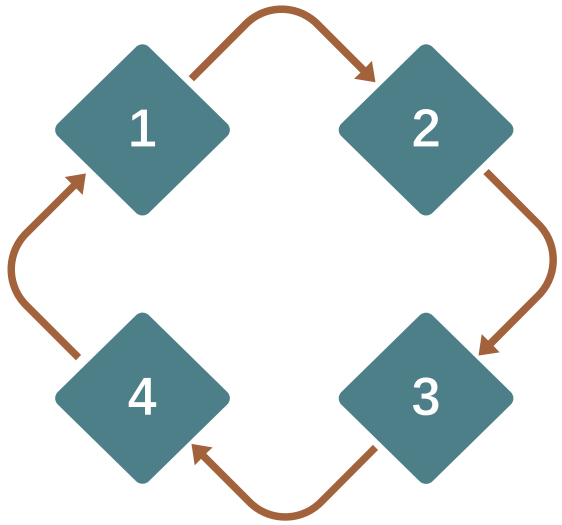
重点：混合运算

从第一个字符开始扫描

在运算的同时对栈的合理性判断继而得出输入的字符串是否是正确合法的

遇到操作符就压入操作符stack中

细节：这个stack我采用的是stack<string>因为毕竟积分符号的长度不止是1



转化为后缀表达式后处理

细节：最后的时候要判断一下栈是不是都变成空了，否则说明输入的不合法

遇到代表多项式的字符串压入多项式stack中

细节：不断读字符直到该字符不是字母为止

04

界面展示

界面展示

```
==========================多项式计算器==========================
1.输入多项式 2.混合运算 3.求逆 4.除法/取模运算 5.求根 6.查询多项式
================================================================
请输入您的选择：1
请输入多项式的长度：3
请按照次数从高到低输入系数：
2 1 0
请输入该多项式的名字：f
您刚刚输入的多项式为：f = 2x^2 + x

输入成功！是否继续输入[y/n]：y


请输入多项式的长度：3
请按照次数从高到低输入系数：
1 -1 2
请输入该多项式的名字：g
您刚刚输入的多项式为：g = x^2 - x + 2

输入成功！是否继续输入[y/n]：
```

界面展示

```
========================多项式计算器========================
1.输入多项式  2.混合运算  3.求逆  4.除法/取模运算  5.求根  6.查询多项式
============================================================
请输入您的选择：6
您想要查询的多项式名称为：f
f = 2x^2 + x
是否继续查询[y/n]：y


您想要查询的多项式名称为：h
抱歉！您想要查询的多项式并没有，请查看是否输入错误！
是否继续查询[y/n]：
```

```
==================多项式计算器==================
1.输入多项式  2.混合运算  6.查询多项式
================================================
请输入您的选择：2
请输入您要计算的表达式：f*f

结果是：4x^2 + 4x + 1
是否继续计算[y/n]：n
```

汇报完毕谢谢观看

- 专业：计算机科学与技术
- 学号：191220177
- 姓名：周俊杰