

多项式计算器实验报告

一.实验目的

熟悉类的操作

二.实验环境

2.1 编程语言和开发工具

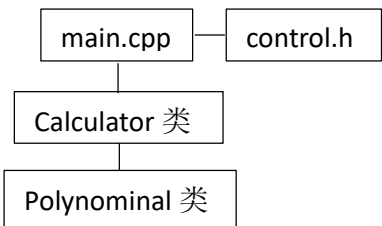
编程语言：ANSI C++

开发工具：Visual Studio 2017 Cummunity

2.2 编码规范

代码自带文档作用（顾名思义）

三.分析与设计



```
> Polynomial Calculator
1 --- Save Polynominal
2 ----- Addition
3 ----- Subtraction
4 ----- Multiplication
5 ----- Differentiation
6 ----- Evaluation
7 ----- Judgment
8 ----- Display Record
9 ----- Reset
0 ----- Exit
Instruction:
```

```
graph LR; subgraph Menu; 1[1 --- Save Polynominal]; 2[2 ----- Addition]; 3[3 ----- Subtraction]; 4[4 ----- Multiplication]; 5[5 ----- Differentiation]; 6[6 ----- Evaluation]; 7[7 ----- Judgment]; 8[8 ----- Display Record]; 9[9 ----- Reset]; 0[0 ----- Exit]; end; Menu --> SR[Save Result]; 8 --> DAP[Delete A Polynominal];
```

3.1 需求分析

利用计算机进行多项式运输

3.2 类结构设计

Calculator 类

```
class Calculator {
public:
    static Calculator* getInstance();
```

```

static void deleteInstance();
void addPoly(const string name, const Polynomial & poly);
void printPoly(const string name);
bool nameExist(const string name);
void display();
bool empty();
Polynomial & getPoly(const string name);
Polynomial left;
Polynomial right;
Polynomial current;
void clear();
void read();
void write();
private:
    map<string, Polynomial> Poly;
    Calculator();
    ~Calculator();
    static Calculator* instance;
    DISALLOW_COPY_AND_ASSIGN(Calculator);
};

```

Polynomial 类

```

class Polynomial {
public:
    Polynomial();
    Polynomial(const Polynomial & source);
    Polynomial(const string input);
    ~Polynomial();
    Polynomial operator+(const Polynomial & right);
    Polynomial operator-(const Polynomial & right);
    Polynomial operator*(const Polynomial & right);
    Polynomial & operator=(const Polynomial & source);
    Polynomial & operator=(const string Input);
    Polynomial differentiate();
    float evaluate(float num);
    bool operator==(const Polynomial & other);
    bool invalid_0();
    bool empty();
    friend ostream & operator<<(ostream & out, const Polynomial & Poly);

    map<int, float> Term;
    void addTerm(int exp, float coef);
};

```

3.3 设计细节

多项式储存识别、多项式与多项式名称的识别、返回主菜单非常方便、部分不回显的清爽

V2.0 新增全局 Esc 键有效

四.代码（关键代码）

```

Calculator *cal = Calculator::getInstance();
void printMenu()
//判断浮点数输入是否合法
bool floatValid(const string Input)
//判断多项式输入是否合法

```

```

bool polyIsValid(const string input)
//判断名称输入是否合法
bool nameIsValid(const string input)
//判断名称是否已存在
bool nameExist(const string input)
//实现随时返回输入字符串
bool inputEsc(string & get)
//菜单功能
void save()
//识别是具体多项式或是名称，并进行合法性判断
int judge(const string target)
//左右多项式的输入
int leftright()
//保存结果多项式
int saveResult()
//菜单功能 + - * == 操作
void operate(const char ch)
//菜单功能
void dif()
//菜单功能
void evl()
//菜单功能
void dsp()
//菜单功能
void res()
//读文件
void read()
//写文件
void write()

```

五.实验结果（截图测试数据及其结果）

```

> Polynomial Calculator > Save a polynomial 1 直接识别一条式子
e.g. p=(1,2)(1,0)
p=(1,2)(1,0)
p = x^2+1 successfully saved!
Press any key to return

```

```

> Polynomial Calculator > Addition 2 可输入多项式或已存名称
>Record:
    p = x^2+1
    q = 5x^6+3x^4
    r = 5x^6+3x^4+x^2+1
Input a concrete polynomial or its name if only existed.
Lvalue:
p
Rvalue:
q
(x^2+1)+(5x^6+3x^4) = 5x^6+3x^4+x^2+1
Input 1 to return, 2 to save the result, other to continue

```

3. 删除多项式子功能

```
> Polynomial Calculator > Display Record  
  
    p = x^2+1  
    q = 5x^6+3x^4  
    r = 5x^6+3x^4+x^2+1  
  
Press D to delete a polynominal, Esc to return
```

```
> Polynomial Calculator > Display Record > Delete A Record  
  
    p = x^2+1  
    q = 5x^6+3x^4  
    r = 5x^6+3x^4+x^2+1  
  
Name:  
p  
  
Delete p = x^2+1 ?  
  
Press Y to delete, Esc to return the menu, others to return.
```

六.心得体会

1. 拷贝构造函数参数必须为 `const` 类型的引用
2. `sscanf` 的用法
3. `getline(cin,string)`要包含`<string>`
4. 文件流 `peek()`指针位置停留在前, `eof()`指针位置同步
5. `c++11`for(auto x: mapName)x.first x.second 的遍历 ; 元素插入 `insert(make_pair(key,value))`

更多内容详见 Readme