实 验 报 告

实验人：王永锋 学号：16337237 日期：2016年6月1日

院（系）：数据科学与计算机学院 专业（班级）：16级计科教务4班

实验题目： 一个多项式计算器的实现

目录

[1 实验目的 1](#_Toc484147313)

[2 实验环境 1](#_Toc484147314)

[2.1 编程语言和开发工具 1](#_Toc484147315)

[2.2 编码规范 2](#_Toc484147316)

[3 实验内容 2](#_Toc484147317)

[4 分析与设计 2](#_Toc484147318)

[4.1 需求分析： 2](#_Toc484147319)

[4.2 结构设计 3](#_Toc484147320)

[4.3 细节设计 4](#_Toc484147321)

[4.3.1 在Polynomial类中， 4](#_Toc484147322)

[4.3.2 在usePoly类中 5](#_Toc484147323)

[5 实验结果 5](#_Toc484147324)

[6 设计心得 6](#_Toc484147325)

# 实验目的

熟悉面向对象的编程思想以及类的使用。

# 实验环境

## 编程语言和开发工具

**开发环境：Ununtu 16.04**

编程语言： c++11

开发工具： vs-code

编译工具：gcc编译器

文件编码：utf-8

## 编码规范

要求遵循良好的程序设计风格来设计和编写程序。基本编码规范：

1. 标识符的命名要到达顾名思义的程度

2. 关键代码提供清晰、准确的注释；

3. 程序版面要求：

a) 不同功能块用空行分隔；

b) 一般一个语句一行；

c) 语句缩进整齐、层次分明。

# 实验内容

# 分析与设计

## 需求分析：

### 需求列表

1. 输入多项式,并使用标识符存储及检索

2. 多项式相加

3. 多项式相减

4. 多项式与常数的乘法运算

5. 求多项式代入某点的值

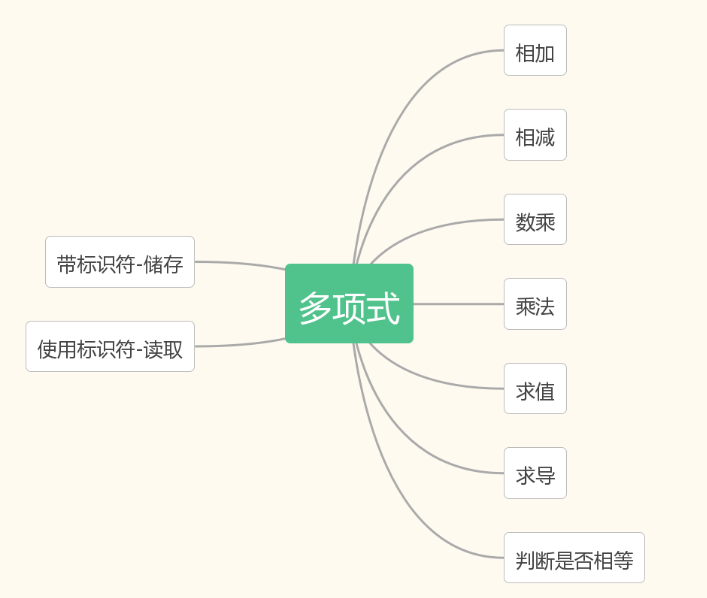
6. 显示所有储存多项式(以降幂形式)

7. 多项式与多项式相乘

8. 判断两个多项式是否相等

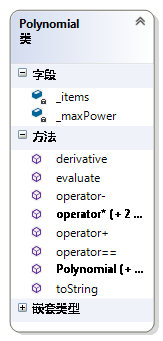
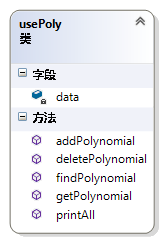
9. 对某个多项式求导

### 系统功能图



## 结构设计

### 类关系图：



聚合关系

## 细节设计

### 在Polynomial类中，

**接口设计**：

//构造函数

Polynomial(int maxPower = MAX\_ITEM\_NUM);

// 生成（maxPower+1）项的多项式，并且初始化指数为对应数字，系数为0

Polynomial(const std::vector<item> t);

// 使用vector来初始化多项式的系数

// 注意此vector的内容需合法

//重载运算符

Polynomial operator+(const Polynomial& rhs) const;//多项式相加

Polynomial operator-(const Polynomial& rhs) const;//多项式相减

Polynomial operator\*(const item& rhs) const;//多项式乘某一项

Polynomial operator\*(const Polynomial& rhs) const;//多项式乘多项式

Polynomial operator\*(coefficient\_t rhs) const;//多项式的数乘

friend Polynomial operator\*(coefficient\_t lhs, const Polynomial& rhs);

//多项式的数乘

bool operator==(const Polynomial& rhs) const;

// 运算接口

double evaluate(coefficient\_t arg) const;// 多项式求值

Polynomial derivative() const; //多项式求导

// I/O

std::string toString() const;//多项式转变成字符串类型输出

friend std::ostream & operator<<(std::ostream& out, const Polynomial & rhs);

//使用cout输出

**数据成员设计**：

// 数据成员如下

power\_t \_maxPower; // 最大次数, +1 即为数组存放的项数

std::vector <item> \_items;// 存放每一项的数组，该数组默认最大项数为MAX\_ITEM\_NUM 为50

// 其中

typedef double coefficient\_t;//系数类型为double

typedef int power\_t;//指数类型为int

struct item{

power\_t power;//每一项的次数

coefficient\_t coefficient;//系数

item(power\_t arg, coefficient\_t arg2):power(arg),coefficient(arg2) {}

bool operator<(const item& rhs) const;

bool operator!=(const item& rhs) const;

};// 每一项

### 在usePoly类中

接口设计：

bool findPolynomial(const std::string& name) const;

//在多项式库中寻找有无该标识符对应的多项式

void addPolynomial(const std::string& name, const Polynomial & poly);

//在多项式库中添加该标识符对应的多项式

void deletePolynomial(const std::string&);

//在多项式库中删除该标识符对应的多项式

Polynomial getPolynomial(const std::string&);

//在该多项式库中取得该标识符对应的多项式

void printAll();

//格式化打印该多项式库中存有的所有多项式

数据成员设计

std::map<std::string, Polynomial> data;

//索引是多项式对应的标识符，内容是该多项式

# 实验结果

设计有代表性的输入数据，并分析测试结果以及针对测试中发现的问题所进行的

修正与改进。

**说明：**

# 设计心得

进行总结，描述所获得的经验和心得体会等。

**注： 提示信息**

1. 可发挥想象力，提高系统的交互体验，可用性和可靠性。

2. 请独立完成，但允许和鼓励相互讨论后自己写程序。由于题目具有开放性，很容易从程序看出是否抄袭。若发现抄袭，将导致抄袭双方或多方大副扣分或直接零分。

**程序测试**

1. 请自行测试类的功能。

2. 界面交互时，请检测用户输入的正确性，保证程序的健壮性。（在用户做出操作后最好打印出提示语句。）

3. 在实验报告中分析测试意图和测试结果。如果通过测试发现程序错误，也分析错误原因、改正方法等，并在实验报告中体现。