

Project 3 实验报告

姓名：王凯祺 学号：16337233 班级：教务 3 班

2017 年 5 月 18 日

1 需求分析

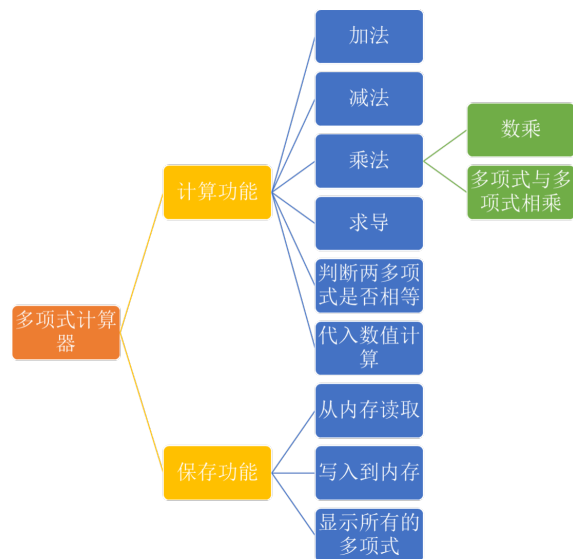
实现一个多项式计算器，支持以下功能：

1. 输入多项式并保存到内存。
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. 显示内存中的多项式。
7. 多项式与多项式相乘。
8. 判断两个多项式是否相等。
9. 对一个多项式求导。

要求：

1. 支持将计算结果保存为变量。
2. 输入的多项式可以为之前保存过的变量。

2 实现思路



2.1 计算功能

创建一个多项式类，用 vector 容器存储从低次到高次的项。

加法：记相加的两个多项式为 a, b ，其次数为 d_a, d_b ，则 $a + b$ 的次数不超过 $\max(d_a, d_b)$ ，对位相加后化简即可。

减法：记减法的两个多项式为 a, b ，其次数为 d_a, d_b ，则 $a + b$ 的次数不超过 $\max(d_a, d_b)$ ，对位相减后化简即可。

数乘：记该多项式为 a ，其次数为 d_a ，则 a 的次数不超过 d_a （不一定是 d_a ，因为与其相乘的常数可能为 0），对位相乘后化简即可。

多项式乘法：记相乘的两个多项式为 a, b ，其次数为 d_a, d_b ，则 $a + b$ 的次数不超过 $d_a + d_b$ ，用 $O(n^2)$ 求一个卷积后化简即可。

求导：记该多项式为 a ，其次数为 d_a ，则 a 的次数不超过 $\max(d_a - 1, 0)$ （不一定是 $d_a - 1$ ，因为与 d_a 可能为 0），简单求个导即可。

判断多项式相等：先判次数，再对每个项分别判一下即可。

2.2 保存功能

用个 map 映射一下，把 string 映射成多项式的类即可。

从 map 中读取：先用 `map::find` 找一下是否有这样的变量，若有就返回该多项式；否则新建一个多项式。

写入到 map：`map[name] = polynomial`。

显示所有的多项式：遍历整个 map。

3 数据设计

一个 vector 就能表示一个多项式了呀:)

```
1 class poly {
2     vector<int> c;
3 };
```

4 类设计和函数设计

函数设计也十分简单啊，对于每个功能重载一个运算符或者写一个函数即可。

```
1 class poly {
2 public:
3     poly();
4     friend istream& operator >> (istream &in, poly &a);
5     friend ostream& operator << (ostream &out, const poly &a);
6     poly operator + (const poly &rhs);
7     poly operator - (const poly &rhs);
8     poly operator * (const int rhs);
9     poly operator * (const poly &rhs);
10    bool operator == (const poly &rhs);
11    void optim(); // 化简多项式
```

```
12     int solve(int x) const; // 代入 x 求值
13     poly derivative(); // 多项式求导
14     int valid(); // 是否为合法的多项式
15 private:
16     vector<int> c;
17 };
```

5 输入与输出

5.1 输入

先输入一个字符串进来，从头扫到尾。

容易找到所有的左括号，从左括号开始遍历到下一个右括号，经过恰好一个逗号和两个数，将这两个数加到对应的项即可。

5.2 输出

从高位往低位输出，遇到 0 就跳过，遇到次数为 1 或 -1、系数为 1 或 -1 的特判一下即可。