项目说明文档

数据结构课程设计

——算数表达式求解

作 者 姓 名： 崔鑫宇

学 号： 1853444

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

# 1 分析

## 1.1 背景分析

本程序为一个模拟计算器的程序，要求用户从键盘上输入中缀算术表达式，包括括号(以等号结尾)，并要求最后计算出表达式的结果，并输出到屏幕上。

## 1.2 功能分析

本程序应该可以对所有输入的表达式进行判断，如果表达式有错能给出适当的提示。本程序支持包括加减、乘除、取余、乘方和括号的操作符，其中优先级是等于<括号<加减<乘除取余<乘方，此外本程序还应该能够处理单目运算符：+或-。

# 2 设计

## 2.1 数据结构设计

如上述描述所述，本程序支持包括加减、乘除、取余、乘方和括号的操作符，而这些操作符之间还应该由优先级的相对高低关系。因此我们需要使用到的是——栈，这一数据结构。

## 

## 2.2 类结构设计

本程序中我们需要使用栈，这一数据结构我们的栈应该具有推入元素、删除尾元素以及提取尾元素的功能。

## 

## 2.3 成员与操作设计

**栈类（MyStack）**

public:

MyStack();

~MyStack() = default;

bool isEmpty();//判断栈是否为空

bool isFull();//判断栈是否为满

bool get\_top(T& x);//提取栈顶元素

bool push(T x);//将元素推入栈

bool pop();//删除栈顶元素

private:

int count;//统计栈中数字

## T data[MAXLEN];//存储数据

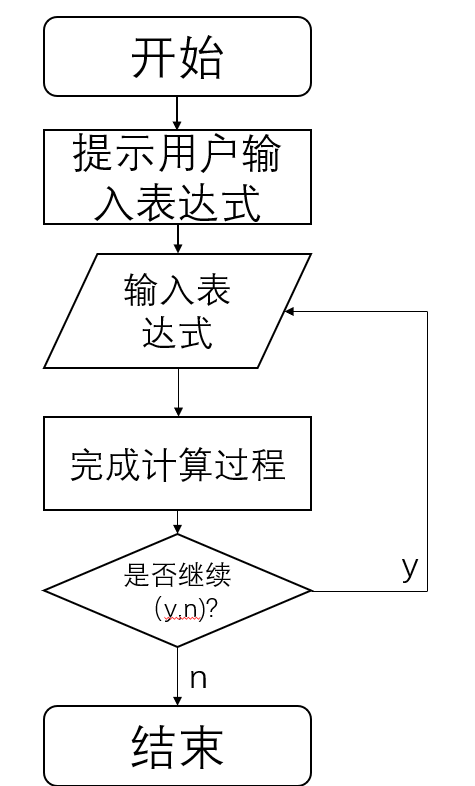
## 2.4 系统设计

系统首先要求用户输入一个表达式（以等号结尾），并完成计算，之后询问用户是否继续进行计算。如果用户输入的是y则重新要求用户输入表达式，如果用户输入的是n则结束本程序。

# 

# 3 实现

### 3.6.1 总体系统流程图



### 3.6.2 总体系统核心代码

if (!isNumber(s[i])) {//检测表达式是否是非法输入

if (priority(s[i]) != -1 && s[i + 1] && !isNumber(s[i + 1]) && priority(s[i + 1]) != -1) {

cerr << "表达式错误！请检查输入";

exit(-1);

}

operate.get\_top(top);

if (priority(s[i]) > priority(top) || s[i] == '(') {

operate.push(s[i]);

}//将高优先级的运算符打入栈内

else {

while (priority(s[i]) <= priority(top)) {

if (top == '#' && s[i] == '=') {

int sum;

operate.pop();

number.get\_top(sum);

cout << sum << endl;

number.pop();

return 0;

}

else if (top == '(' && s[i] == ')') {

++i;

}//当左右括号相遇时，跳过右括号，删除左括号

else {

number.get\_top(a);

number.pop();

number.get\_top(b);

number.pop();

}

if (top == '+') {

b += a;

number.push(b);

}

else if (top == '-') {

b -= a;

number.push(b);

}

else if (top == '\*') {

b \*= a;

number.push(b);

}

else if (top == '/') {

b /= a;

number.push(b);

}

else if (top == '%') {

b %= a;

number.push(b);

}

else if (top == '^') {

b = pow(b, a);

number.push(b);

}

operate.pop();

//取前一个运算符，并与当前运算符进行比较

operate.get\_top(top);

}

//将优先级高的运算符实现计算

operate.push(s[i]);

}

### }

}

string expression;

char ch = 'y';

while (ch == 'y') {

cout << "输入表达式:" << endl;

cin >> expression;

if (!isNumber(expression[0]) && expression[0] != '+' && expression[0] != '-') {

cerr << "首位元素应该为数字或正负号!";

exit(-1);

}

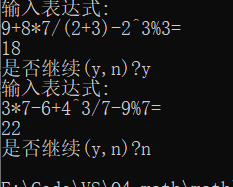
calculate(expression);

cout << "是否继续(y,n)?";

cin >> ch;

}

### 3.6.3 总体系统截屏示例



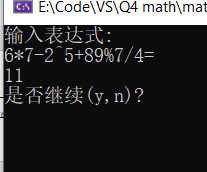
# 4 测试

## 4.1 正常运算功能测试

**测试用例(1)：**6\*7-2^5+89%7/4=

**预期结果：**11

**实验结果：**

****

**测试用例(2)：**(2+7)^4+1821\*3-12921/16=

**预期结果：**11217

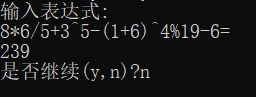
**实验结果：**



**测试用例(3)：**8\*6/5+3^5-(1+6)^4%19-6=

**预期结果：**239

**实验结果：**



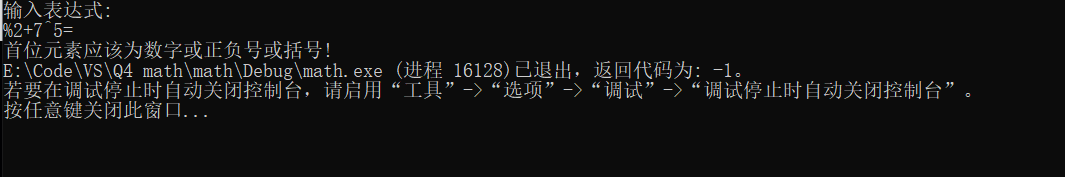
## 4.2不良输入测试

### 4.2.1 首元素为不良输入

**测试用例：**%2+7^5=

**预期结果：**提示首元素应该为数字或正负号以及括号，程序立即结束。

**实验结果：**



### 4.2.2 两个运算符相邻且它们中没有括号

**测试用例：**2^5%4++2=

**预期结果：**提示表达式中内容非法，程序立即结束。

**实验结果：**

