**项目说明文档**

**数据结构课程设计**

**——算术表达式求解**

作者姓名：香宁雨

学号：1954098

指导教师：张颖

学院、专业：软件学院 软件工程

同济大学

Tongji University

目录

[1.功能分析 2](#_Toc58626521)

[2.设计 3](#_Toc58626522)

[2.1数据结构设计 3](#_Toc58626523)

[2.2类结构设计 3](#_Toc58626524)

[2.3类的成员与方法 4](#_Toc58626525)

[2.4测试函数 6](#_Toc58626526)

[3.实现 6](#_Toc58626527)

[3.1element结构体 6](#_Toc58626528)

[3.2suffix(string s)函数 7](#_Toc58626529)

[3.3print(vector<element>test)函数 10](#_Toc58626530)

[3.4calculate(string s) 11](#_Toc58626531)

[3.5run(double left, double right, char op)函数 12](#_Toc58626532)

[3.6判断函数：isOperationSymbol(char symbol)函数、isNumber(char num)函数、isParenthsisMatching(string s)函数、check(string s)函数 13](#_Toc58626533)

[3.7符号优先级：priority(char symbol) 15](#_Toc58626534)

[4.输出测试 15](#_Toc58626535)

[4.1题目测试输出 15](#_Toc58626536)

[4.2特性输出 16](#_Toc58626537)

# 1.功能分析

在平时的运算过程中我们经常会遇到过于复杂的算术表达式，但人计算会出差错，但电脑在正确的逻辑下不会出差错，所以本项目实现了输入表达式输出计算结果的目标，实现了简单计算器的功能。读入从键盘上输入的中缀算术表达式，包括括号，计算出表达式的值。程序对所有输入的表达式作简单的判断，如果表达式有错误，抛出“invalid argument: Wrong input”的错误。

程序可以计算=、()、+、-、\*、/、%、^以及单目运算符+和-。

# 2.设计

## 2.1数据结构设计

对于计算机而言，相比于中缀表达式，后缀表达式和前缀表达式更易进行计算，也就是一般常称的波兰表达式和逆波兰表达式，而计算机进行计算时，通常采用栈这种数据结构来进行压栈及计算。

本项目采用逆波兰表达式进行算数表达式的计算，逆波兰表达式中不含‘（‘及’）‘，将后缀表达式的符号压入栈中，遇到数字时将符号出栈进行计算，实现计算中缀表达式的值的功能。

## 2.2类结构设计

栈可以由链表这种基础数据结构来实现，实现后进先出，指只从最后进行数据的插入和弹出，为简化实现难度，本项目中的链表结构采用双向链表的方式来存储所有元素，使链表类元素既可以向前遍历也可以向后遍历，简化遍历的难度。

## 2.3类的成员与方法

#### 2.3.1链表类（ListNode、List、Iter）



#### 2.3.2栈类（Stack）



2.4测试函数  


# 3.实现

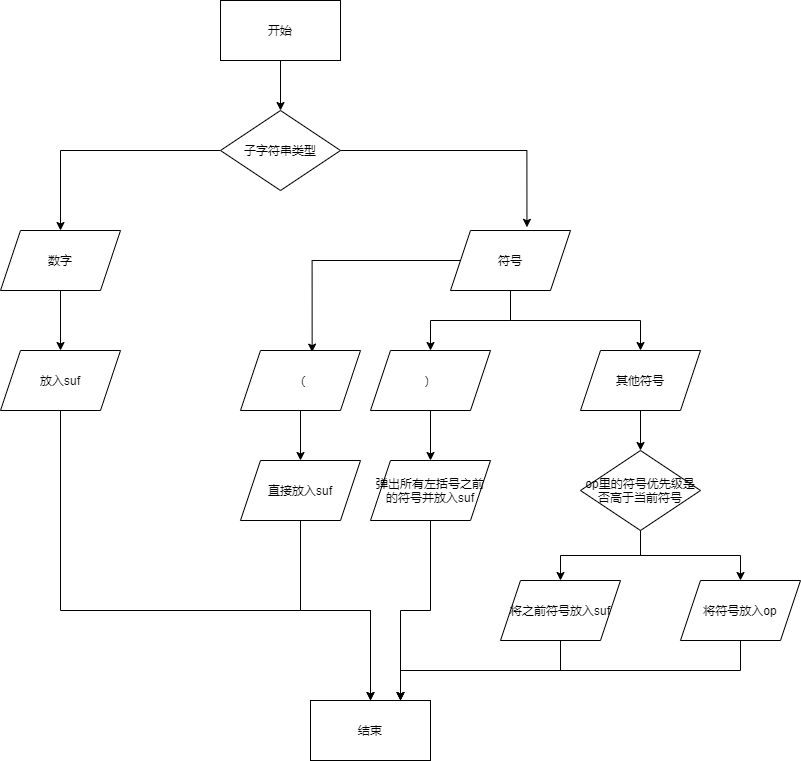
## 3.1element结构体

我们通过element结构体来进行数字和符号的存储，为之后栈处理进行准备。



## 3.2suffix(string s)函数

#### 3.2.1流程图



#### 3.2.2代码实现





## 3.3print(vector<element>test)函数

此函数为调试过程中的测试函数。



## 3.4calculate(string s)

#### 3.4.1流程图



#### 3.4.2代码实现



## 3.5run(double left, double right, char op)函数

通过run函数来实现各个运算方法。

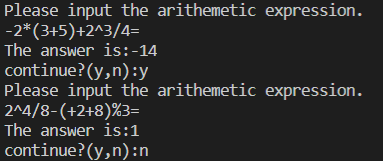


3.6判断函数：isOperationSymbol(char symbol)函数、isNumber(char num)函数、isParenthsisMatching(string s)函数、check(string s)函数

3.7符号优先级：priority(char symbol)

# 4.输出测试

## 4.1题目测试输出



## 4.2特性输出

