数据结构与算法

实验报告

实验题目：实验二题目一——实现栈或队列（实现栈）

姓 名：刘帅

学 号：2020212267

日 期：2021年4月30日

自我评分：【 B+ 】

自我评分说明：A+，A，B+，B，B-，C，D，分别对应分数95、90、85、80、75、70、60

诚信声明

本人郑重承诺：本实验程序和实验报告均是本人独立学习和工作所获得的成果。尽我所知，实验报告中除特别标注的地方外，不包含其他同学已经发表或撰写过的成果；实验程序中对代码工作的任何帮助者所作的贡献均做了明确的说明，并表达了谢意。

如有抄袭，本人原因承担因此而造成的任何后果。

特此声明。

签名：刘帅

日期：2021年4月30日

程序引用说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 引用项 | 来源 | 相同代码行数 |
| 1 | 栈的构造及功能 | 课本 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 小计 | | |  |

总代码行数64; 引用占比\_\_\_\_\_\_60%\_\_\_\_\_\_\_\_\_\_

1、实验简介

本实验为栈的构造实验。

2、程序框架

1. 与构造相关的函数：

linkstack() { top = 0; }//构造空栈

2．与操作有关函数：

void push(T x);//入栈

T pop();//出栈

T Gettop();//查找栈顶元素

3、判断函数

bool isempty() { return(top == 0) ? true : false;}//判断链表是否为空

3、关键代码实现

【哪些函数是你认为最能体现自己工作成果的函数，说明函数实现基本思想（可用文字或图表示），以及具体的实验步骤（用伪代码或带注释代码）】

屏幕上有字

描述已自动生成3.1void linkstack<T>::push(T x) {//入栈

3.2 T linkstack<T>::pop() //出栈

图形用户界面, 文本

描述已自动生成

3.3 T linkstack<T>::Gettop() //查找栈顶元素文本

描述已自动生成

文本

描述已自动生成3.4 析构函数

4、不足

无

5、心得体会

书里的基础内容，了解了栈的存储结构及其特性有助于后续实验的结构选择。

数据结构与算法

实验报告

实验题目：扩展线性表题目5应用实验（简易计算器）

姓 名：刘帅

学 号：2020212267

日 期：2021年4月30日

自我评分：【 A+ 】

自我评分说明：A+，A，B+，B，B-，C，D，分别对应分数95、90、85、80、75、70、60

诚信声明

本人郑重承诺：本实验程序和实验报告均是本人独立学习和工作所获得的成果。尽我所知，实验报告中除特别标注的地方外，不包含其他同学已经发表或撰写过的成果；实验程序中对代码工作的任何帮助者所作的贡献均做了明确的说明，并表达了谢意。

如有抄袭，本人原因承担因此而造成的任何后果。

特此声明。

签名：刘帅

日期：2021年4月30日

程序引用说明

序号 引用项 来源 相同代码行数

1 无

小计

总代码行数112; 引用占比\_\_\_\_\_\_0%\_\_\_\_\_\_\_\_\_\_

1、实验简介

本实验为栈的扩展实验，利用栈的特性将输入表达式转化为后缀表达式，并实现计算器的加减功能。

2、程序框架

1. 与操作有关函数：

（1）Void convert()//将中缀表达式转化为后缀表达式

（2）double cal()//计算

2、与存储有关的函数：

（1）void mapset() //利用散列给不同操作符赋值

3、关键代码实现

【哪些函数是你认为最能体现自己工作成果的函数，说明函数实现基本思想（可用文字或图表示），以及具体的实验步骤（用伪代码或带注释代码）】

一、Void convert()//将中缀表达式转化为后缀表达式

说明：结点结构体中包括三个信息，操作数、操作符及指示符（bool型，指示该结点所存为数还是符），在循环开始，将三者置为零，从左至右依次进栈，若为操作数，则直接压入后缀表达式栈，若操作数是多位相连，采用temp.num = temp.num \* 10 + str[i] - '0'，主要思想是将多个字符存成一个数，压入栈。例：12+35，对于“12”，通过算式将12压入栈，而不是将1，2依次压入栈。对操作符部分，则需分类讨论，遇到左括号，直接压入操作符栈，遇到右括号，则将操作符栈中的操作符依次弹出，直至遇到左括号。对于“+-\*/”四个操作符，若操作符优先级小于等于操作符栈顶，则将操作符栈的元素持续压入到后缀表达式栈中，直至低优先级操作符进栈。当操作结束，将操作符栈的操作符全部压入后缀表达式栈，遇到左右括号则直接弹出，不进后缀表达式栈。此时，所有操作符及操作数全部进后缀表达式栈，且栈底元素为中缀表达式最左侧元素。

二、double cal()//后缀表达式的计算

说明：首先建立tep栈将后缀表达式倒置，即后缀表达式从左至右依次计算。遇到操作数，则先将其压入操作符栈，遇到操作符，将操作符栈栈顶的两元素弹出，与操作符进行运算，并将运算后数值压入操作符栈。

三、mapset（）；//利用散列给不同运算符赋权重

4、不足

1、对于后缀表达式的操作，可以设计成队列，从而避免了建立新栈倒置表达式的过程。

2、可以思考其他计算器的功能以及实际计算器加减功能的实现原理。

5、心得体会

本实验的精髓在于首先将中缀表达式转化为后缀表达式（逆波兰式），实验前在洛谷上遇到过几次这样的问题，为我提供了思路，才得以完成实验。