山东大学 软件 学院

数据结构 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201900302030 | 姓名：邵嘉明 | | 班级： 数字媒体技术 |
| 实验题目：实验四 堆栈的应用 | | | |
| 实验学时：2 | | 实验日期： 2020.11.8 | |
| 实验目的：掌握堆栈的使用。 | | | |
| 硬件环境：PC | | | |
| 软件环境：Visual Studio 2019 | | | |
| 实验步骤与内容：   1. 输入一个数学表达式（假定表达式输入格式合法），计算表达式结果并输出。 2. 数学表达式由单个数字和运算符“+”、“-”、“\*”、“/”、“(、) ”构成，例如 2 + 3 \* ( 4 + 5 ) - 6 / 4。 3. 变量、输出采用整数，只舍不入。 | | | |
| 结论分析与体会：  对堆栈的概念与应用有了更深的理解与体会。 | | | |

**本次实验主要代码：**

#include<iostream>

#include<stack>

#include<string>

using namespace std;

int symbolPriority(int calcstate, char inputStr)

{

int result=0;

switch (inputStr) {

case '+':

case '-':

result = 1;

break;

case '\*':

case '/':

result = 2;

break;

case '(':

if (calcstate == 0)

result = 3;

else

result = 0;

break;

case '#':

result = 0;

break;

default:

break;

}

return result;

}

int calculate(char symbol, int firstNumber, int secondNumber)

{

int result=0;

switch (symbol) {

case '+':

result = firstNumber + secondNumber;

break;

case '-':

result = firstNumber - secondNumber;

break;

case '\*':

result = firstNumber \* secondNumber;

break;

case '/':

result = firstNumber / secondNumber;

break;

default:

break;

}

return result;

}

int main(int argc, char\* argv[]){

cout << "Input" << endl;

string inputStr;

cin >> inputStr;

stack<char> calcSymbol;

stack<int> opeObeject;

calcSymbol.push('#');

string number;

for (int i = 0; i < inputStr.length(); i++) {

if (isdigit(inputStr[i])) {

while (isdigit(inputStr[i]) || inputStr[i] == '.') {

number.push\_back(inputStr[i]);

i++;

}

int res = atof(number.c\_str());

opeObeject.push(res);

number.clear();

i--;

}

else if (inputStr[i] == '+' || inputStr[i] == '-' || inputStr[i] == '\*' || inputStr[i] == '/' || inputStr[i] == '(') {

if (symbolPriority(0, inputStr[i]) > symbolPriority(1, calcSymbol.top()))

calcSymbol.push(inputStr[i]);

else {

while (symbolPriority(0, inputStr[i]) <= symbolPriority(1, calcSymbol.top())) {

char temp = calcSymbol.top();

calcSymbol.pop();

int secondNumber = opeObeject.top();

opeObeject.pop();

int firstNumber = opeObeject.top();

opeObeject.pop();

opeObeject.push(calculate(temp, firstNumber, secondNumber));

}

calcSymbol.push(inputStr[i]);

}

}

else if (inputStr[i] == ')') {

while (calcSymbol.top() != '(') {

char temp = calcSymbol.top();

calcSymbol.pop();

int secondNumber = opeObeject.top();

opeObeject.pop();

int firstNumber = opeObeject.top();

opeObeject.pop();

opeObeject.push(calculate(temp, firstNumber, secondNumber));

}

calcSymbol.pop();

}

else {

cout << "inputStr is not valid!" << endl;

return 0;

}

}

while (calcSymbol.top() != '#') {

char temp = calcSymbol.top();

calcSymbol.pop();

int secondNumber = opeObeject.top();

opeObeject.pop();

int firstNumber = opeObeject.top();

opeObeject.pop();

opeObeject.push(calculate(temp, firstNumber, secondNumber));

}

cout << "Output" << endl;

cout << opeObeject.top() << endl;

cout << "End";

system("pause");

return 0;

}