山东大学 软件 学院

数据结构 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号： | 姓名： | | 班级： |
| 实验题目：堆栈 | | | |
| 实验学时：2小时 | | 实验日期：2018.11.4 | |
| 实验目的：  掌握堆栈的使用。 | | | |
| 硬件环境： 笔记本电脑 | | | |
| 软件环境：Win10+Vistual Studio 2017 | | | |
| 实验步骤与内容：  实验内容：  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。  实验步骤：  numeration.h  #pragma once  int numeration(char op, int op1, int op2)  {  int result;  switch (op)  {  case '+':  result = op1 + op2;  break;  case '-':  result = op1 - op2;  break;  case '\*':  result = op1 \* op2;  break;  case '/':  result = op1 / op2;  break;  default:  break;  }  return result;  }  sequence.h  #pragma once  int sequence(int state, char a)  {  int result;  switch (a)  {  case '+':  case '-':  result = 1;  break;  case '\*':  case '/':  result = 2;  break;  case '(':  if (state == 0)  result = 3;  else  result = 0;  break;  case '#':  result = 0;  break;  default:  break;  }  return result;  }  源.cpp  #include<iostream>  #include<stack>  #include<string>  #include"numeration.h"  #include"sequence.h"  using namespace std;  int main()  {  string x;  cout << "Input" << endl;  while (cin >> x)  {  stack<char> operate;  stack<int> number;  operate.push('#');  string figure;  for (int i = 0; i < x.length(); i++)  {  if (isdigit(x[i]))  {  while (isdigit(x[i]) || x[i] == '.')  {  figure.push\_back(x[i]);  i++;  }  int a = atoi(figure.c\_str());  number.push(a);  figure.clear();  i--;  }  else if (x[i] == '+' || x[i] == '-' || x[i] == '\*' || x[i] == '/' || x[i] == '(')  {  if (sequence(0, x[i]) > sequence(1, operate.top()))  operate.push(x[i]);  else {  while (sequence(0, x[i]) <= sequence(1, operate.top()))  {  char t = operate.top();  operate.pop();  int operand2 = number.top();  number.pop();  int operand1 = number.top();  number.pop();  number.push(numeration(t, operand1, operand2));  }  operate.push(x[i]);  }  }  else if (x[i] == ')')  {  while (operate.top() != '(')  {  char t = operate.top();  operate.pop();  int operand2 = number.top();  number.pop();  int operand1 = number.top();  number.pop();  number.push(numeration(t, operand1, operand2));  }  operate.pop();  }  else  {  cout << "error!" << endl;  return 0;  }  }  while (operate.top() != '#')  {  char t = operate.top();  operate.pop();  int operand2 = number.top();  number.pop();  int operand1 = number.top();  number.pop();  number.push(numeration(t, operand1, operand2));  }  cout << "Output" << endl;  cout << number.top() << endl;  cout << "End" << endl;  }  system("pause");  return 0;  } | | | |
| 结论分析与体会：  运行结果： | | | |