# МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №4**

**по дисциплине «Программирование» Тема: «Динамические структуры данных»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 9303 |  | Жорже М.А. |
| Преподаватель |  | Чайка К. В. |

Санкт-Петербург 2020

**Цель работы.**

Изучать принципы работы структуры данных стек, реализовать его при помощи инструментария языка С++.

**Задание.**

**Вариант 1 Стековая машина.**

Требуется написать программу, которая последовательно выполняет подаваемые ей на вход арифметические операции над числами с помощью стека на базе **массива.**

1. Реализовать **класс** CustomStack, который будет содержать перечисленные ниже методы. Стек должен иметь возможность хранить и работать с типом данных ***int***

Объявление класса стека:

class CustomStack {

public:

// методы push, pop, size, empty, top + конструкторы, деструктор

private:

// поля класса, к которым не должно быть доступа извне

protected: // в этом блоке должен быть указатель на массив данных

int\* mData;

};

Перечень методов класса стека, которые должны быть реализованы:

* + **void push(int val)** - добавляет новый элемент в стек
  + **void pop()** - удаляет из стека последний элемент
  + **int top()** - доступ к верхнему элементу
  + **size\_t size()** - возвращает количество элементов в стеке
  + **bool empty()** - проверяет отсутствие элементов в стеке
  + **extend(int n)** - расширяет исходный массив на n ячеек

1. Обеспечить в программе считывание из потока ***stdin*** последовательности (не более 100 элементов) из чисел и арифметических операций (+, -, \*, / (деление нацело)) разделенных пробелом, которые программа должна интерпретировать и выполнить по следующим правилам:
   * Если очередной элемент входной последовательности - число, то положить его в стек
   * Если очередной элемент - знак операции, то применить эту операцию над двумя верхними элементами стека, а результат положить обратно в стек (следует считать, что левый операнд выражения лежит в стеке глубже)
   * Если входная последовательность закончилась, то вывести результат (число в стеке)
   * Если в процессе вычисления возникает ошибка:
   * например вызов метода **pop** или **top** при пустом стеке (для операции в стеке не хватает аргументов)
   * по завершении работы программы в стеке более одного элемента
   * программа должна вывести "**error**" и завершиться.

**ПРИЛОЖЕНИЕ**

**А ИСХОДНЫЙ КОД ПРОГРАММЫ**

#include<iostream>

#include<string.h>

#include<stdio.h>

#define LEN 50

using namespace std;

class CostumStack {

protected:

int\* mDate;

private:

int sizes;

int sizestack;

int sizestr;

char s[LEN][LEN];

int checkSymb(char\* st){

if(strcmp(st, "\*") == 0 || strcmp(st, "-") == 0 || strcmp(st, "/") == 0 || strcmp(st, "+") == 0)

return 1;

return -1;

}

double Operate(char symb, int x, int y){

switch(symb){

case '+':

return x + y;

case '-':

return x - y;

case '\*':

return x \* y;

}

return x / y;

}

public:

CostumStack(){

sizes = 0;

sizestr = 0;

sizestack = 100;

mDate = new int[sizestack];

}

~CostumStack(){

delete [] mDate;

}

size\_t size(){

return sizes;

}

bool empty(){

if(sizes != 1)

return 1;

return 0;

}

void push(int val){

mDate[sizes] = val;

sizes++;

}

void pop(int k){

for(int i = k; i <= sizes; i++)

mDate[i] = mDate[i+1];

sizes--;

}

int top(){

if(this->empty()){

this->~CostumStack();

cout << "error" << endl;

exit(0);

}

return mDate[sizes-1];

}

void Initial(char \*str){

int k = 0;

for(int i = 0; i <= strlen(str); i++){

if(str[i] == ' ' || str[i] == '\0'){

s[sizestr++][k] = '\0';

k = 0;

}else {

s[sizestr][k++] = str[i];

}

}

k = 0;

for (int i = 0; i < sizestr; i++){

if(strlen(s[i]) > 1)

push(atoi(s[i]));

else

if(checkSymb(s[i]) < 0)

push(atoi(s[i]));

}

}

void Main(){

int dn = 0, k = 0;

for(int i = 0; i <= sizestr; i++){

if(checkSymb(s[i]) > 0){

if(dn >= 2){

//cout << mDate[k-2] << s[i][0] << mDate[k-1] << " = ";

mDate[k-1] = Operate(s[i][0], mDate[k-2], mDate[k-1]);

//cout << mDate[k-1] << endl;

pop(k-2); dn--;k--;

}else{

this->~CostumStack();

cout << "error" << endl;

exit(0);

}

}else{

dn++; k++;

}

}

}

};

int main(){

CostumStack Stack;

char symb;

char \*str = new char[LEN];

int i = 0;

while((symb = getchar()) != EOF && symb != '\n'){

str[i++] = symb;

}

Stack.Initial(str);

Stack.Main();

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

//cout << "size = " << Stack.size() << endl;

return 0;

}