**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №3**

**по дисциплине «Алгоритмы и структуры данных»**

**Тема: Стек, дек и очередь**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6383 |  | Лавренкова Е.Л. |
| Преподаватель |  | Шолохова О.М. |

Санкт-Петербург

2017

**Цель работы.**

Практическое применение и освоение стека и очереди, структур данных, функций для представления информации на языке программирования С++.

**Задание (вариант №11.а).**

Вычислить как целое число значение выражения (без переменных), записанного в постфиксной форме в заданном текстовом файле postfix.

**Основные теоретические положения.**

Постфиксной (префиксной) формой записи выражения aDb называется запись, в которой знак операции размещен за (перед) операндами: abD (Dab). Эта запись удобна тем, что она не требует скобок. Для перевода числа в префиксную форму записи выражения мы используем инфиксную форму записи. Пример: инфиксная запись - a+b^c^d\*e, соответственно, префиксная - abc^d^e\*+ .

**Спецификация программы.**

*Назначение программы*.

Программа предназначена для вычисления значения выражения, записанного в постфиксной форме как целого числа.

*Описание программы*.

Программа написана на языке C++. Входные данные – файл postfix, с записанным в него выражением. Пример ввода:

122 15 3 5 - + /

Выходные данные – вычисленное целое число. Пример вывода:

Результат = 9.

*Реализация*.

Представляем стек с помощью класса:

class Stack {

private:

struct node;

node \*topOfStack;

public:

Stack()

{

topOfStack = NULL;

};

base top(void);

void pop(void);

base pop2(void);

void push(const base &x);

bool isNull(void);

void destroy(void);

};

В исходном классе указан список свойств и методов для использования внутри класса (private):

* Структура node
* Указатель на вершину стека типа node

В исходном классе указан список методов, доступных другим функциям и объектам программы(public):

* Основные функции
* Указатель на вершину стека = 0

Исходная структура Stack:

struct Stack::node

{

base \*hd;

node \*tl;

node()

{

hd = NULL; tl = NULL;

}

};

Структура содержит переменную типа base-hd, где base может быть любым типом данных, tl - указатель на следующий элемент типа node,а также node() инициализатор - при создании объекта присвоит NULL hd and tl.

Функции:

* base Stack::top(void)

Функция для получения верхнего элемента стека. Возвращаемое значение: значение верхнего элемента.

* void Stack::pop(void)

Функция для удаления верхнего элемента стека. Входные и выходные данные отсутствуют.

* base Stack::pop2(void)

Функция для возвращения значения верхнего элемента, для последующего его удаления. Входные данные отсутствуют.

* void Stack::push(const base &x)

Функция для добавления в конец стека элемента со значением x. Входные данные: элемент со значением x. Выходные данные отсутствуют.

* bool Stack::isNull(void)

Функция для проверки стека на пустоту. Входные данные отсутствуют. Возвращаемое значение: true при пустом стеке, false при непустом стеке.

* Stack::~Stack(void)

Деструктор. Входные и выходные данные отсутствуют.

**Тестирование.**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Исходные данные** | **Инфиксная запись** | **Результат** |
| 1 | 10 4 \* 3 2 + / | 10\*4/(3+2) | 8 |
| 2 | 45 8 / 7 \* | 45/8\*7 | 35 |
| 3 | 100 9 / 7 / | 100/9/7 | 1 |
| 4 | 100 9 7 / / | 100/(9/7) | 100 |

**Выводы.**

В процессе выполнения работы были получены практические навыки по применению классов, стека и очереди, структур данных на языке программирования С++.

**Приложение. Исходный код.**

***Файл lr3head.h***

#pragma once

namespace st\_modul1

{

typedef int base;

class Stack {

private:

struct node;

node \*topOfStack;

public:

Stack()

{

topOfStack = NULL;

};

base top(void);

void pop(void);

base pop2(void);

void push(const base &x);

bool isNull(void);

~Stack();

};

}

***Файл lr3main.cpp***

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <windows.h>

#include "lr3head.h"

#include <string>

using namespace std;

using namespace st\_modul1;

int main() {

char a[100];

setlocale(0, "Rus");

cout << "Ввод строки с постфиксной записью выражения." << endl;

ifstream fin("postfix.txt");

fin >> noskipws; // включить манипулятор

if (!fin)

{

cout << "Файл отсутствует!\n";

system("Pause");

return 1;

}

int n = 0;

fin.getline(a, 100);

n = strlen(a)-1;

cout << "Длина строки = " << n << endl;

cout << a;

cout << endl;

cout << "Вычислить!" << endl;

Stack s;

for (int i = 0; i < n; i++)

{

cout << "Шаг\_: " << i + 1 << " символ = " << a[i] << endl; // dem

if (a[i] == '+')

s.push(s.pop2() + s.pop2());

if (a[i] == '-')

{

base tmp = s.pop2();

s.push(s.pop2() - tmp);

}

if (a[i] == '\*')

s.push(s.pop2() \* s.pop2());

if (a[i] == '/')

{

base tmp = s.pop2();

s.push(s.pop2() / tmp);

}

if (a[i] == '^')

{

base tmp = s.pop2();

s.push((base)pow(s.pop2(), tmp));

}

if ((a[i] >= '0') && (a[i] <= '9'))

s.push(0);

while ((a[i] >= '0') && (a[i] <= '9'))

{

s.push(10 \* s.pop2() + (a[i++] - '0'));

cout << "Шаг\_: " << i + 1 << " символ = " << a[i] << endl;// dem

}

}

cout << "Результат = " << s.pop2() << endl;

system("Pause");

return 0;

}

***Файл lr3func.cpp***

#include <iostream>

#include <cstdlib>

#include "lr3head.h"

using namespace std;

namespace st\_modul1

{

struct Stack::node

{

base \*hd;

node \*tl;

node()

{

hd = NULL; tl = NULL;

}

};

base Stack::top(void)

{

if (topOfStack == NULL) { cerr << "Error: top(null) \n"; exit(1); }

else return \*topOfStack->hd;

}

void Stack::pop(void)

{

if (topOfStack == NULL) { cerr << "Error: pop(null) \n"; exit(1); }

else

{

node \*oldTop = topOfStack;

topOfStack = topOfStack->tl;

delete oldTop->hd;

delete oldTop;

}

}

base Stack::pop2(void)

{

if (topOfStack == NULL)

{

cerr << "Error: pop(null) \n";

exit(1);

}

else

{

node \*oldTop = topOfStack;

base r = \*topOfStack->hd;

topOfStack = topOfStack->tl;

delete oldTop->hd;

delete oldTop;

return r;

}

}

void Stack::push(const base &x)

{

node \*p;

p = topOfStack;

topOfStack = new node;

if (topOfStack != NULL) {

topOfStack->hd = new base;

\*topOfStack->hd = x;

//cout << "push -> " << x << endl;

topOfStack->tl = p;

}

else { cerr << "Memory not enough\n"; exit(1); }

}

bool Stack::isNull(void)

{

return (topOfStack == NULL);

}

Stack::~Stack(void)

{

while (topOfStack != NULL) {

pop();

}

}

}