# Учреждение образования

# «Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Отчёт по лабораторной работе №7

Вариант 2

Выполнил:

студент 1 курса 2 группы Самцевич Алексей

Минск 2020

Задание №1: Разработать функцию, которая удаляет из стека первый отрицательный элемент, если такой есть.

Код:

Главный файл:

//

// main.cpp

// laba\_7.1

//

// Created by Alex Samtsevich on 18.05.2020.

// Copyright © 2020 Alex Samtsevich. All rights reserved.

//

#include <iostream>

#include "functions.hpp"

using namespace std;

int main() {

setlocale(LC\_ALL, "Rus");

int choice;

Stack \*myStk = new Stack; //выделение памяти для стека

myStk->head = NULL; //инициализация первого элемента

for (;;) {

cout << "Выберите команду:" << endl;

cout << "1 - Добавление элемента в стек" << endl;

cout << "2 - Извлечение элемента из стека" << endl;

cout << "3 - Вывод стека" << endl;

cout << "4 - Удаление элемента" << endl;

cout << "5 - Выход" << endl;

cin >> choice;

switch (choice) {

case 1: cout << "Введите элемент: " << endl;

cin >> choice;

push(choice, myStk);

break;

case 2: choice = pop(myStk);

if (choice != -1)

cout << "Извлеченный элемент: " << choice << endl;

break;

case 3: cout << "Весь стек: " << endl;

show(myStk);

break;

case 4:

elemDelete(myStk);

break;

case 5:

return 0;

break;

}

}

return 0;

}

Заголовочный файл:

//

// functions.hpp

// laba\_7.1

//

// Created by Alex Samtsevich on 18.05.2020.

// Copyright © 2020 Alex Samtsevich. All rights reserved.

//

#ifndef functions\_hpp

#define functions\_hpp

#include <stdio.h>

struct Stack {

int data; //информационный элемент

Stack \*head; //вершина стека

Stack \*next; //указатель на следующий элемент

};

void show(Stack \*myStk); //прототип

int pop(Stack \*myStk); //прототип

void push(int x, Stack \*myStk); //прототип

void elemDelete(Stack \*myStk);

#endif /\* functions\_hpp \*/

Файл с функциями:

//

// functions.cpp

// laba\_7.1

//

// Created by Alex Samtsevich on 18.05.2020.

// Copyright © 2020 Alex Samtsevich. All rights reserved.

//

#include "functions.hpp"

#include <iostream>

#include "functions.hpp"

using namespace std;

void push(int x, Stack \*myStk) {

Stack\* e = new Stack; //выделение памяти для нового элемента

e->data = x; //запись элемента x в поле v

e->next = myStk->head; //перенос вершины на следующий элемент

myStk->head = e; //сдвиг вершины на позицию вперед

}

int pop(Stack \*myStk) {

if (myStk->head == NULL) {

cout << "Стек пуст!" << endl;

return -1; //если стек пуст - возврат -1

}

else {

Stack \*e = myStk->head; //е - переменная для хранения адреса элемента

int a = myStk->head->data; //запись числа из поля data в переменную a

myStk->head = myStk->head->next; //перенос вершины

delete e; //удаление временной переменной

return a; //возврат значения удаляемого элемента

}

}

void show(Stack \*myStk) {

Stack\* e = myStk->head;

int a;

if (e == NULL) {

cout << "Стек пуст!" << endl;

}

while (e != NULL) {

a = e->data; //запись значения в переменную a

cout << a << " ";

e = e->next;

}

cout << endl;

}

void elemDelete(Stack \*myStk) {

Stack\* e = myStk->head;

Stack\* prev = NULL;

while (e != NULL) {

if (e->data < 0) {

if (e == myStk->head) {

myStk->head = e->next;

free(e);

e->data = NULL;

e->next = NULL;

break;

}

else {

prev->next = e->next;

free(e);

e->data = NULL;

e->next = NULL;

}

}

prev = e;

e = e->next;

}

}

Скриншот:

