БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра вычислительных методов и программирования

ОТЧЕТ

по лабораторной работе

**«Динамическая структура СТЕК»**

|  |  |  |
| --- | --- | --- |
| Выполнила |  | Семенова В. С. |
| Проверила |  | Шатилова О. О. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Минск 2023

**Цель:** изучить алгоритмы работы с динамическими структурами данных в виде стека.

# Общая постановка задачи

Написать программу по созданию, добавлению, просмотру и решению приведенных далее для однонаправленного линейного списка типа Stack.

**Вариант 11**

В созданном списке вычислить среднее арифметическое и заменить им первый элемент.

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <cmath>

#include <random>

#include <string>

#include <windows.h>

using namespace std;

struct Stack {

double info;

Stack\* next;

};

Stack\* InStack(Stack\*, int);

void View(Stack\*);

void Del\_All(Stack\*\*);

Stack\* InStack(Stack\* p, int in) {

Stack\* t = new Stack;

t->info = in;

t->next = p;

return t;

}

void View(Stack\* t) {

while (t != NULL) {

cout << " " << t->info << endl;

t = t->next;

}

}

void Del\_All(Stack\*\* p) {

Stack\* t;

while (\*p != NULL) {

t = \*p;

\*p = (\*p)->next;

delete t;

}

}

void Task(Stack\* p) {

int n = 0; double sr, sum = 0;

Stack\* t = p;

while (t != NULL) {

sum+=t->info;

n++;

t = t->next;

}

sr = sum / n;

cout << "Среднее арифметическое = " << sr;

t = p;

while (t->next->next != NULL) {

t = t->next;

}

Stack\* m = new Stack;

m->info = sr;

m->next = NULL;

t->next = m;

}

void main()

{

Stack \* beginn=NULL;

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

int i,in, n, kod;

while (true) {

cout << "\nСоздать - 1" << endl;

cout << "Добавить - 2" << endl;

cout << "Просмотр - 3" << endl;

cout << "Очистить память - 4" << endl;

cout << "Вычислить среднее арифметическое и заменить им первый элемент. - 5" << endl;

cout << "Выход - 0"<<endl;

cin >> kod;

switch(kod){

case 1: case 2:

if (kod == 1 && beginn != NULL) {

cout << "Очистите память!" << endl;

break;

}

cout << "Введите кол-во элементов = "; cin >> n;

for (i = 1; i <= n; i++) {

cin >> in;

beginn = InStack(beginn, in);

}

if (kod == 1) cout << "Выполнено\n";

break;

case 3:

if (!beginn) {

cout << "Стек пуст!" << endl;

break;

}

cout << "--- Стек ---" << endl;

View(beginn);

break;

case 4: Del\_All(&beginn);

cout << "Память очищена!" << endl;

break;

case 5:

Task(beginn);

break;

case 0: break;

if (beginn != NULL)

Del\_All(&beginn);

cout << "Конец!";

return;

default: cout << "Ошибка ввода!" << endl;

}

}

}