БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра вычислительных методов и программирования

ОТЧЕТ

по лабораторной работе

**«Динамическая структура Очередь»**

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

Минск 2023

**Цель:** изучить возможности работы со списками, организованными в виде очереди.

# Общая постановка задачи

Написать программу по созданию, добавлению (в начало, в конец), просмотру (с начала, с конца) и решению приведенной в подразделе 3.3 задачи для двунаправленных линейных списков.

**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 Spis2 {

double info;

Spis2\* next, \* prev;

};

//---------------------------------------------------------------------------

void Create\_Spis2(Spis2\*\*, Spis2\*\*, double);

void Add\_Spis1(Spis2\*\*, double);

void Add\_Spis2(Spis2\*\*, double);

void View\_Spis1(Spis2\*);

void View\_Spis2(Spis2\*);

void Del\_All(Spis2\*\*, Spis2\*\*);

void Task(Spis2\*\*);

//------------------- Создание первого элемента -----------------------------

void Create\_Spis2(Spis2\*\* b, Spis2\*\* e, double in) {

Spis2\* t = new Spis2;

t->info = in;

t->next = t->prev = NULL;

\*b = \*e = t;

}

void Del\_All(Spis2\*\* p, Spis2\*\* e) {

Spis2\* t;

while (\*p != NULL) {

t = \*p;

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

delete t;

}

\*e = NULL;

};

//------------------- Добавление элемента в список --------------------------

void Add\_Spis1(Spis2\*\* b, double in) {

Spis2\* t = new Spis2;

t->info = in;

t->prev = NULL;

t->next = \*b;

(\*b)->prev = t;

\*b = t;

}

void Add\_Spis2(Spis2\*\* e, double in) {

Spis2\* t = new Spis2;

t->info = in;

t->next = NULL;

t->prev = \*e;

(\*e)->next = t;

\*e = t;

}

//--------------------- Просмотр элементов списка ---------------------------

void View\_Spis1( Spis2\* t) {

while (t != NULL) {

cout << t->info << endl;

t = t->next;

}

}

void View\_Spis2( Spis2\* t) {

while (t != NULL) {

cout << t->info << endl;

t = t->prev;

}

}

void Task(Spis2\*\* p) {

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

Spis2\* t = \*p;

while (t != NULL) {

sum += t->info;

n++;

t = t->next;

}

sr = sum / n;

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

(\*p)->info = sr;

}

void main()

{

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

int kod, kod1; double in; Spis2\* begin = NULL, \* end = NULL, \* t = NULL;

char Str[2][10] = { "начало ", "конец " };

while (true) {

cout << "\n\tСоздать - 1\n\tДобавить - 2\n\tПросмотр - 3\n\tОчистить память - 4\n\tЗадание - 5\n\tВыход - 0 : ";

cin >> kod;

switch (kod) {

case 1: if (begin != NULL) {

cout << "Очистите память!" << endl;

break;

}

cout << "Введите первый элемент = "; cin >> in;

Create\_Spis2(&begin, &end, in);

cout << "Начало создано = " << begin->info << endl;

break;

case 2:

if (!begin) {

cout << "Создайте первый элемент!"; break;

}

cout << "Элемент = "; cin >> in;

cout << "Добавить в начало - 0, Добавить в конец - 1 : "; cin >> kod1;

if (kod1 == 0) { Add\_Spis1(&begin, in); t = begin; }

else { Add\_Spis2(&end, in); t = end; }

cout << "Добавлено в " << Str[kod1] << " " << t->info << endl;

break;

case 3: if (!begin) {

cout << "Очередь пуста!" << endl;

break;

}

cout << "Просмотреть с начала - 0, Просмотреть с конца - 1:";

cin >> kod1;

if (kod1 == 0) {

t = begin;

cout << "-- Начало --" << endl;

View\_Spis1(t);

}

else {

t = end;

cout << "--- Конец --" << endl;

View\_Spis2(t);

}

break;

case 4:

Del\_All(&begin, &end);

cout << "Память очищена!" << endl;

break;

case 5:

if (!begin) {

cout << "Очередь пуста!" << endl;

break;

}

Task(&begin); break;

case 0: if (begin != NULL)

Del\_All(&begin,&end);

return;

}

}

}