МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра информационно-аналитических систем безопасности

имени Л.С. Берштейна

**ОТЧЕТ**

**По Лабораторной работе № 9**

***«Динамическое распределение памяти, списки»***

**Вариант - 20**

Выполнил:

студент гр. КТбо1-10

Мельман Д.Э

Проверил:

Профессор кафедры ИАСБ

Барковский С.А.

«\_\_\_» 2020 г.

Таганрог – 2021

Код Программы:

#include <iostream>

#include <list>

#include <iterator>

#include <windows.h>

#include <string>

using namespace std;

struct notebook {

string data;

string time;

string meeting;

string importance;

};

list<notebook> notebookList; // <тип данных> <имя контейнера>

notebook April; // <значение>

list <notebook> ::iterator it; // <контейнер><его тип> ::iterator <имя инератора>

// 1 - Добавление данных в начало списка

void GetData\_front(list<notebook>& v) {

cout << endl;

cout << "Дата: ";

getline(cin, April.data);

cout << "Время: ";

getline(cin, April.time);

cout << " Место: ";

getline(cin, April.meeting);

cout << "Важность: ";

getline(cin, April.importance);

v.push\_front(April); // Толкать вперёд

system("cls");

}

// 2 - Добавление данных в конец списка

void GetData\_back(list<notebook>& v) {

cout << endl;

cout << "Дата: ";

getline(cin, April.data);

cout << "Время: ";

getline(cin, April.time);

cout << " Место: ";

getline(cin, April.meeting);

cout << "Важность: ";

getline(cin, April.importance);

v.emplace\_back(April); // Поставить обратно

system("cls");

}

// 5 - Удалить элемент по порядковому номеру

void Delete\_Number(list<notebook>& v) {

it = notebookList.begin(); // С начала

int num;

cout << "Введите номер элемента: ";

cin >> num;

cin.ignore();

advance(it, num - 1); // Продвигать итератор

notebookList.erase(it); // Cтереть

system("cls");

}

// 6 - Удалить элемент по заданной информации

void Delete\_by(list<notebook>& v) {

string text;

int num;

cout << "По какому критерию искать?"<< endl

<< "1. Дата" << endl

<< "2. Время" << endl

<< "3. Место " << endl

<< "4. Важность" << endl;

cin >> num;

cin.ignore();

switch (num) {

case 1:

cout << "Введите Дату: ";

getline(cin, text);

for (auto& b : v) {

if (b.data == text) {

v.remove(b); // Удалить

}

}

break;

case 2:

cout << "Введите Время: ";

getline(cin, text);

for (auto& b : v) {

if (b.time == text) {

v.remove(b); // Удалить

}

}

break;

case 3:

cout << "Введите Место: ";

getline(cin, text);

for (auto& b : v) {

if (b.meeting == text) {

v.remove(b); // Удалить

}

}

break;

case 4:

cout << "Введите Важность: ";

getline(cin, text);

for (auto& b : v) {

if (b.importance == text) {

v.remove(b); // Удалить

}

}

break;

}

system("cls");

}

// 7 - Добавить элемент на заданное место списка

void ADD\_Number(list<notebook>& v) {

it = notebookList.begin(); // С начала

int num;

cout << "Введите номер элемента: ";

cin >> num;

cout << endl;

advance(it, num - 1); // Продвигать итератор

cout << "Дата: ";

getline(cin, April.data);

cout << endl;

cout << "Время: ";

getline(cin, April.time);

cout << endl;

cout << "Место: ";

getline(cin, April.meeting);

cout << endl;

cout << "Важность: ";

getline(cin, April.importance);

v.insert(it, April); // Вставить

system("cls");

}

// 8 - Добавить элемент в середину списка

void ADD\_Middle(list<notebook>& v) {

it = notebookList.begin(); // С начала

int size = notebookList.size();

advance(it, size / 2);

cout << endl;

cout << "Дата: ";

getline(cin, April.data);

cout << "Время: ";

getline(cin, April.time);

cout << "Место: ";

getline(cin, April.meeting);

cout << "Важность: ";

getline(cin, April.importance);

v.insert(it, April); // Вставить <позиция> <значение>

system("cls");

}

// 9 - Удалить элемент из середины списка

void Delete\_Middle(list<notebook>& v) {

it = notebookList.begin(); // С начала

int size = notebookList.size();

int half = size / 2;

advance(it, half);

notebookList.erase(it); // Стереть

system("cls");

}

// 10 - Вывод элементов на экран

void Output\_Data(const list<notebook>& v) {

cout << endl << "\t\t\t" << "Список Дел" << endl;

for (const auto& April : v) {

cout << endl;

cout << "Дата: " << April.data << "\t"

<< "Время: " << April.time << "\t"

<< "Место: " << April.meeting << "\t"

<< "Важность: " << April.importance << endl;

}

system("pause");

system("cls");

}

int main() {

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

while (1) {

int count;

cout << "Действие:" << endl

<< "1. Добавить элемент в начало списка" << endl

<< "2. Добавить элемент в конец списка" << endl

<< "3. Удалить элемент из начала списка" << endl

<< "4. Удалить элемент из конца списка" << endl

<< "5. Удалить элемент по порядковому номеру" << endl

<< "6. Удалить элемент по заданной информации" << endl

<< "7. Добавить элемент на заданное место списка" << endl

<< "8. Добавить элемент в середину списка" << endl

<< "9. Удалить элемент из середины списка" << endl

<< "10. Вывести элементы списка" << endl << endl;

cin >> count;

cin.ignore();

switch (count) {

case 1:

system("cls");

GetData\_front(notebookList); // Добавление элемента в начало списка

break;

case 2:

system("cls");

GetData\_back(notebookList); // Добавление элемента в конец списка

break;

case 3:

system("cls");

notebookList.pop\_front(); // Удаление элемента из начала списка

break;

case 4:

system("cls");

notebookList.pop\_back(); // Удалить элемент из конца списка

break;

case 5:

system("cls");

Delete\_Number(notebookList); // Удалить элемент по порядковому номеру

break;

case 6:

system("cls");

Delete\_by(notebookList); // Удалить элемент по заданной информации

break;

case 7:

system("cls");

ADD\_Number(notebookList); // Добавить элемент на заданное место списка

break;

case 8:

system("cls");

ADD\_Middle(notebookList); // Добавить элемент в середину списка

break;

case 9:

system("cls");

Delete\_Middle(notebookList); // Удалить элемент из середины списка

break;

case 10:

system("cls");

Output\_Data(notebookList); // Вывести элементы списка

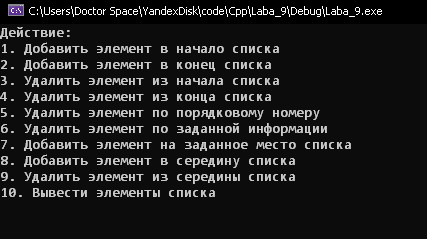
break;

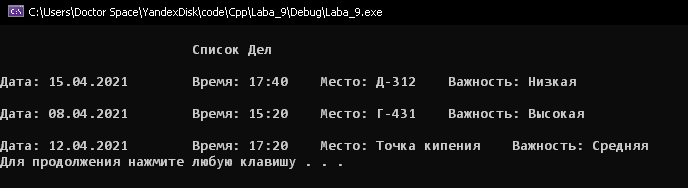
}

}

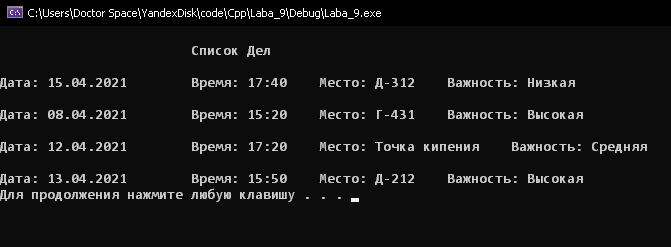
}

Пример:

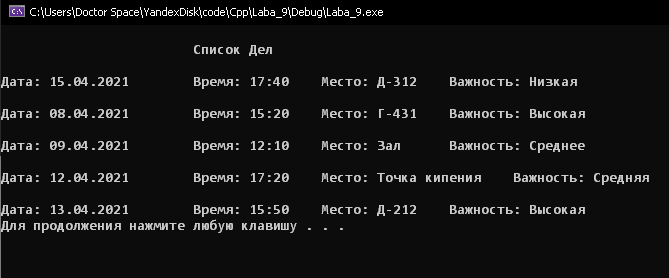
1. Добавляем элементы (через 1 пункт)
2. После добавления некоторых элементов выводим элементы (через 10 пункт)



1. Добавим элимент в конец списка (через 2 пункт и вывидем на экран)



1. Добавим элемент в середину списка (через 8 пункт и вывидем на экран)



1. А теперь удалим начальный элемент списка (через 3пункт и вывидем на экран)

