# Учреждение образования

# «Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Отчёт по лабораторной работе №9

Вариант 2

Выполнил:

студент 1 курса 2 группы Самцевич Алексей

Минск 2020

Задание №1: **deleteKFirst(int k)** – функция удаления **К** первых элементов списка.

Код:

//

// main.cpp

// laba\_9.1

//

// Created by Alex Samtsevich on 19.05.2020.

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

//

#include <iostream>

#include <fstream>

#include <stdio.h>

using namespace std;

const unsigned int NAME\_SIZE = 30;

const unsigned int CITY\_SIZE = 20;

struct Address {

char name[NAME\_SIZE];

char city[CITY\_SIZE];

Address \*next;

Address \*prev;

};

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

int menu(void) {

char s[80]; int c;

cout << endl;

cout << "1 - Ввод имени" << endl;

cout << "2 - Удаление имени" << endl;

cout << "3 - Вывод на экран" << endl;

cout << "4 - Поиск" << endl;

cout << "5 - Удаление К первых элементов" << endl;

cout << "7 - Выход" << endl;

cout << endl;

do {

cout << "Ваш выбор: ";

cin.sync();

gets(s);

cout << endl;

c = atoi(s);

} while (c < 0 || c > 6);

return c;

}

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

void insert(Address \*e, Address \*\*phead, Address \*\*plast) {

Address \*p = \*plast;

if (\*plast == NULL) {

e->next = NULL;

e->prev = NULL;

\*plast = e;

\*phead = e;

return;

}

else {

p->next = e;

e->next = NULL;

e->prev = p;

\*plast = e;

}

}

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

Address\* setElement() {

Address\* temp = new Address();

if (!temp) {

cerr << "Ошибка выделения памяти памяти";

return NULL;

}

cout << "Введите имя: ";

cin.getline(temp->name, NAME\_SIZE-1,'\n');

cin.ignore(cin.rdbuf()->in\_avail());

cin.clear();

cout << "Введите город: ";

cin.getline(temp->city, CITY\_SIZE-1,'\n');

cin.ignore(cin.rdbuf()->in\_avail());

cin.clear();

temp->next = NULL;

temp->prev = NULL;

return temp;

}

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

void outputList(Address \*\*phead, Address \*\*plast) {

Address \*t = \*phead;

while (t) {

cout << t->name << ' ' << t->city << endl;

t = t->next;

}

cout << "" << endl;

}

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

void find(char name[NAME\_SIZE], Address \*\*phead) {

Address \*t = \*phead;

while (t) {

if (!strcmp(name, t->name)) break;

t = t->next;

}

if (!t) {

cerr << "Имя не найдено" << endl;

}

else {

cout << t->name << ' ' << t->city << endl;

}

}

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

void delet(char name[NAME\_SIZE], Address \*\*phead, Address \*\*plast) {

struct Address \*t = \*phead;

while (t) {

if (!strcmp(name, t->name)) break;

t = t->next;

}

if (!t) {

cerr << "Имя не найдено" << endl;

}

else {

if (\*phead == t) {

\*phead = t->next;

if (\*phead) {

(\*phead)->prev = NULL;

}

else {

\*plast = NULL;

}

}

else {

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

if (t != \*plast) {

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

}

else {

\*plast = t->prev;

}

}

delete t;

cout << "Элемент удален" << endl;

}

}

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

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

int deleteKFirst(int k, Address \*head) {

int i = 0;

while (i < k) {

Address \*temp = head->next;

delete head;

head = temp;

i++;

}

return 0;

}

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

int main(void) {

Address \*head = NULL;

Address \*last = NULL;

setlocale(LC\_CTYPE, "Rus");

while(true) {

switch (menu()) {

case 1:

insert(setElement(), &head, &last);

break;

case 2: {

char dname[NAME\_SIZE];

cout << "Введите имя: ";

cin.getline(dname, NAME\_SIZE-1,'\n');

cin.ignore(cin.rdbuf()->in\_avail());

cin.sync();

delet(dname, &head, &last);

}

break;

case 3:

outputList(&head, &last);

break;

case 4: {

char fname[NAME\_SIZE];

cout << "Введите имя: ";

cin.getline(fname, NAME\_SIZE - 1, '\n');

cin.ignore(cin.rdbuf()->in\_avail());

cin.sync();

find(fname, &head);

}

break;

case 5: {

int k;

cout << "Сколько элементов вы хотите удалить? ";

cin >> k;

deleteKFirst(k, head);

}

break;

case 7:

exit(0);

default:

exit(1);

}

}

return 0;

}

//void writeToFile(Address \*\*phead) {

// struct Address \*t = \*phead;

// FILE \*fp;

// errno\_t err = fopen(&fp, "mlist", "wb");

// if (err) {

// cerr << "Файл не открывается" << endl;

// exit(1);

// }

// cout << "Сохранение в файл" << endl;

// while (t) {

// fwrite(t, sizeof(struct Address), 1, fp);

// t = t->next;

// }

// fclose(fp);

//}

//

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

//

//void readFromFile(Address \*\*phead, Address \*\*plast) {

// struct Address \*t;

// FILE \*fp;

// errno\_t err = fopen\_s(&fp, "mlist", "rb");

// if (err) {

// cerr << "Файл не открывается" << endl;

// exit(1);

// }

// while (\*phead) {

// \*plast = (\*phead)->next;

// delete \*phead;

// \*phead = \*plast;

// }

// \*phead = \*plast = NULL;

// cout << "Загрузка из файла" << endl;

// while (!feof(fp)) {

// t = new Address();

// if (!t) {

// cerr << "Ошибка выделения памяти" << endl;

// return;

// }

// if (1 != fread(t, sizeof(struct Address), 1, fp)) {

// break;

// }

// insert(t, phead, plast);

// }

// fclose(fp);

//}

Скриншот:

