Беларускі дзяржаўны тэхналагічны ўніверсітэт

Факультэт інфармацыйных тэхналогій

Кафедра праграмнай інжэнерыі

Лабараторная работа №9

Па дысцыпліне «Асновы алгарытмізацыі і праграмавання»

На тэму «Двухсувязные спісы»

Выканала:

Студэнтка 1 курса 6 группы

Жучкевіч Кацярына Сяргееўна

Выкладчык: асс. Андронава М.В.

Мінск, 2024



#include <iostream>

#include <fstream>

#include <Windows.h>

using namespace std;

const unsigned int NAMEMAX = 30;

const unsigned int CITYNAMEMAX = 20;

struct Address

{

char name[NAMEMAX];

char city[CITYNAMEMAX];

Address\* next;

Address\* prev;

};

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

int menu(void)

{

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

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. Выход" << endl;

cout << "6. Пошук элемента з найбольшым імем" << endl;

cout << endl;

do {

cout << "Ваш выбор: ";

cin.sync();

gets\_s(s);

cout << endl;

c = atoi(s);

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

return c;

}

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

void insert(Address\* e, Address\*\* phead, Address\*\* plast) //Добавление в конец списка

{

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() // Создание элемента и ввод его значений с клавиатуры

{

Address\* temp = new Address();

if (!temp)

{

cerr << "Ошибка выделения памяти памяти";

return NULL;

}

cout << "Введите имя: ";

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

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

cin.clear();

cout << "Введите город: ";

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

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

cin.clear();

temp->next = NULL;

temp->prev = NULL;

return temp;

}

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

void outputList(Address\*\* phead, Address\*\* plast) //Вывод списка на экран

{

Address\* t = \*phead;

while (t)

{

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

t = t->next;

}

cout << "" << endl;

}

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

void find(char name[NAMEMAX], Address\*\* phead) // Поиск имени в списке

{

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[NAMEMAX], Address\*\* phead, Address\*\* plast) // Удаление имени

{

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;

}

}

void findMax(Address\*\* phead) {

Address\* temp = \*phead;//задаем пачатковае значэнне для буфернага элемента

Address\* max = \*phead;//задаем пачатковае значэнне для мах элемента

temp = temp->next;

while (temp) {

if (strlen(temp->name) > strlen(max->name))//шукаем мах элемент

max = temp;

temp = temp->next;

}

cout << max->name << " " << max->city << endl;//выводзім яго

}

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

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[NAMEMAX];

cout << "Введите имя: ";

cin.getline(dname, NAMEMAX - 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[NAMEMAX];

cout << "Введите имя: ";

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

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

cin.sync();

find(fname, &head);

}

break;

case 5: exit(0);

case 6: findMax(&head);

default: exit(1);

}

}

return 0;

}

