Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №12**

Дисциплина: «Основы теории алгоритмов и структуры данных»

Тема: Алгоритмы поиска. Работа со списками. Метод интерполяционного поиска

Вариант 14

Выполнил:

Студент группы РИС-20-2б

Вичугов Алексей Дмитриевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цели работы**

Получить практические навыки сортировки массивов.

**Постановка задачи**

1. Создать динамический массив из записей (в соответствии с вариантом), содержащий не ме-нее 100 элементов. Для заполнения элементов массива использовать ДСЧ.

2. Предусмотреть сохранение массива в файл и загрузку массива из файла.

3. Предусмотреть возможность добавления и удаления элементов из массива (файла).

4. Выполнить поиск элемента в массиве по ключу в соответствии с вариантом. Для поиска ис-пользовать метод Бойера-Мура и интерполяционный метод.

**Анализ задачи**

1. Для решения задачи необходимо:
   1. Сформировать массив структур и заполнить его:   
      struct Massiv{

string fio;

int d,m,g;

string adres;

};

Massiv \*m;

Void Create(){};

Void CUser(){};

* 1. Создать функцию, производящую поиск элемента линейным методом:  
     void LineMetod(){

string key;

bool nai=false;

cout << "Запущен линейный метод поиска. " << endl;

cout << "Адрес человека, которого необходимо найти: "; getline(cin, key);getline(cin, key);

for (int i=0;i<r;i++){

if (key==m[i].adres){

cout << "Элемент номер " << i+1 << ": " << m[i].fio << "; "; if (m[i].d<10) cout << "0"; cout << m[i].d << "."; if (m[i].m<10) cout << "0"; cout << m[i].m << "." << m[i].g << "; " << m[i].adres << endl;

nai=true;

}

}

if (nai==false) cout << "Элемент не найден. " << endl;

}

* 1. Создать функцию, производящую поиск элемента методом прямого поиска:

void Rovniy(){

string kucha,getted;

int e=0,nomber=0,dlin=0,sovpalo;

bool nai=false;

for (int i=0;i<r;i++) {kucha+="\n";kucha+=m[i].adres;}

cout << "Адрес искомого элемента: ";

cin.ignore(30,'\n');

getline(cin,getted);

for (int i=0;i<kucha.length();i++){

dlin++;

if (kucha[i]=='\n') nomber++;

if (getted[e++]==kucha[i]) {

sovpalo++;

}

else {e=0;sovpalo=0;}

if (sovpalo==getted.length()&&sovpalo==dlin){

nai=true;

sovpalo=0;

e=0;

cout << "Элемент, содержащий введённые данные: " << nomber << ". " << m[nomber-1].fio << "; "; if (m[nomber-1].d<10) cout << "0"; cout << m[nomber-1].d << "."; if (m[nomber-1].m<10) cout << "0"; cout << m[nomber-1].m << "." << m[nomber-1].g << "; " << m[nomber-1].adres << endl;

}

if (kucha[i]=='\n') dlin=0;

}

if (!nai) cout << "Элемент не найден. " << endl;

}

* 1. Создать функцию, производящую поиск элемента интерполяционным методом:

void Interpolar(){

SortiData();

string key;

int middle, left=0, right=r-1, ztchk=0,curdat=0;

bool nai=false;

cout << "Запупщен интерполяционный метод поиска. " << endl;

cout << "Дата рождения человека, которого необходимо найти: " << endl << "День: "; cin >> ztchk;

while (ztchk<1||ztchk>31||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать 31! Повторите ввод: ";

cin >> ztchk;

}

curdat=ztchk;

cout << "Месяц рождения элемента: ";

cin >> ztchk;

while (ztchk<1||ztchk>12||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать 12! Повторите ввод: ";

cin >> ztchk;

}

curdat+=ztchk\*100;

cout << "Год рождения элемента: ";

cin >> ztchk;

while (ztchk<1900||ztchk>2021||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть больше 1900 и меньше 2021! Повторите ввод: ";

cin >> ztchk;

}

curdat+=ztchk\*10000;

while (!nai&&left!=right){

middle=left+((curdat-(m[left].d+m[left].m\*100+m[left].g\*10000))\*(right-left))/((m[right].d+m[right].m\*100+m[right].g\*10000)-(m[left].d+m[left].m\*100+m[left].g\*10000));

if (m[middle].d+m[middle].m\*100+m[middle].g\*10000==curdat) {

cout << "Элемент номер " << middle+1 << ": " << m[middle].fio << "; "; if (m[middle].d<10) cout << "0"; cout << m[middle].d << "."; if (m[middle].m<10) cout << "0"; cout << m[middle].m << "." << m[middle].g << "; " << m[middle].adres << ";" << endl;

nai=true;

}

else{

if (m[middle].d+m[middle].m\*100+m[middle].g\*10000<curdat) left=middle+1;

else right = middle-1;

}

}

if (!nai) cout << "Элемент не найден. " << endl;

}

1. В ходе работы был использованы типы данных:
   1. int:  
      int n,ran,v=-1,r;
   2. Bool:

bool mc=false,nmc=false;

* 1. String:  
      string adres; string fio;

1. Данные для решения задачи представлены в виде:
   1. Целочисленных переменных:  
      int itd=0,itu=0;
   2. Строчных переменных:  
      string key; string adr;
2. В ходе работы программы использовалась структура Massiv:  
   struct Massiv{  
   string fio;  
   int d,m,g;  
   string adres;  
   };
3. Ввод значений производился с помощью оператора cin и функции getline:  
   cin >> v;  
   getline(cin,getted);
4. При решении поставленных задач использовались циклы:
   1. Для заполнения массива элементов и поиска определённого элемента цикл for:  
      for (int i=0;i<r;i++){}
   2. Для проверки правильности ввода цикл while:

while (ztchk<1900||ztchk>2021||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть больше 1900 и меньше 2021! Повторите ввод: ";

**Полный код программы**

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <ctime>

#include <string>

#include <iomanip>

using namespace std;

struct Massiv{

string fio;

int d,m,g;

string adres;

};

Massiv \*m;

int n,ran,v=-1,r;

bool mc=false,nmc=false;

void Create(){

ofstream obaza("baza.txt");

srand(time(0));

cout << "Количество элементов: ";

cin >> r;

m = new Massiv[r];

obaza << "Количество элементов: " << r << endl;

for (int i=0;i<r;i++){

n=rand()%10;

ran=rand()%10;

m[i].d=1+rand()%28;

m[i].m=1+rand()%12;

m[i].g=1900+rand()%121;

switch (n){

case 0: m[i].fio="Анастасия "; break;

case 1: m[i].fio="Мария "; break;

case 2: m[i].fio="Дарья "; break;

case 3: m[i].fio="Анна "; break;

case 4: m[i].fio="Елизавета "; break;

case 5: m[i].fio="Александр "; break;

case 6: m[i].fio="Максим "; break;

case 7: m[i].fio="Иван "; break;

case 8: m[i].fio="Артём "; break;

case 9: m[i].fio="Дмиртий "; break;

}

switch (ran){

case 0: if (n<5) m[i].fio+="Владимировна "; else m[i].fio+="Владимирович "; break;

case 1: if (n<5) m[i].fio+="Ивановна "; else m[i].fio+="Иванович "; break;

case 2: if (n<5) m[i].fio+="Александровна "; else m[i].fio+="Александрович "; break;

case 3: if (n<5) m[i].fio+="Сергеевна "; else m[i].fio+="Сергеевич "; break;

case 4: if (n<5) m[i].fio+="Антоновна "; else m[i].fio+="Антонович "; break;

case 5: if (n<5) m[i].fio+="Петровна "; else m[i].fio+="Петрович "; break;

case 6: if (n<5) m[i].fio+="Дмитриевна "; else m[i].fio+="Дмитриевич "; break;

case 7: if (n<5) m[i].fio+="Анатольевна "; else m[i].fio+="Анатольевич "; break;

case 8: if (n<5) m[i].fio+="Артёмовна "; else m[i].fio+="Артёмович "; break;

case 9: if (n<5) m[i].fio+="Семёновна "; else m[i].fio+="Семёнович "; break;

}

ran=rand()%10;

switch (ran){

case 0: if (n<5) m[i].fio+="Иванова"; else m[i].fio+="Иванов"; break;

case 1: if (n<5) m[i].fio+="Петрова"; else m[i].fio+="Петров"; break;

case 2: if (n<5) m[i].fio+="Сидорова"; else m[i].fio+="Сидоров"; break;

case 3: if (n<5) m[i].fio+="Кузнецова"; else m[i].fio+="Кузнецов"; break;

case 4: if (n<5) m[i].fio+="Попова"; else m[i].fio+="Попов"; break;

case 5: if (n<5) m[i].fio+="Смирнова"; else m[i].fio+="Смирнов"; break;

case 6: if (n<5) m[i].fio+="Васильева"; else m[i].fio+="Васильев"; break;

case 7: if (n<5) m[i].fio+="Соколова"; else m[i].fio+="Соколов"; break;

case 8: if (n<5) m[i].fio+="Новикова"; else m[i].fio+="Новиков"; break;

case 9: if (n<5) m[i].fio+="Волкова"; else m[i].fio+="Волков"; break;

}

ran=rand()%10;

switch (ran){

case 0: m[i].adres="Пермь, "; break;

case 1: m[i].adres="Москва, "; break;

case 2: m[i].adres="Санкт-Петербург, "; break;

case 3: m[i].adres="Екатеринбург, "; break;

case 4: m[i].adres="Новосибирск, "; break;

case 5: m[i].adres="Новокузнецк, "; break;

case 6: m[i].adres="Новороссийск, "; break;

case 7: m[i].adres="Омск, "; break;

case 8: m[i].adres="Архангельск, "; break;

case 9: m[i].adres="Курган, "; break;

}

ran=rand()%10;

switch(ran){

case 0: m[i].adres+="ул. Екатерининская, "; break;

case 1: m[i].adres+="ул. Мостовая, "; break;

case 2: m[i].adres+="ул. Центральная, "; break;

case 3: m[i].adres+="ул. Королёва, "; break;

case 4: m[i].adres+="ул. Мира, "; break;

case 5: m[i].adres+="ул. Луговая, "; break;

case 6: m[i].adres+="ул. Лаврова, "; break;

case 7: m[i].adres+="ул. Институтская, "; break;

case 8: m[i].adres+="ул. Арбат, "; break;

case 9: m[i].adres+="ул. Революции, "; break;

}

ran=rand()%10;

switch(ran){

case 0: m[i].adres+="д. 17"; break;

case 1: m[i].adres+="д. 14"; break;

case 2: m[i].adres+="д. 46"; break;

case 3: m[i].adres+="д. 8"; break;

case 4: m[i].adres+="д. 5-А"; break;

case 5: m[i].adres+="д. 91"; break;

case 6: m[i].adres+="д. 194"; break;

case 7: m[i].adres+="д. 18"; break;

case 8: m[i].adres+="д. 7"; break;

case 9: m[i].adres+="д. 81"; break;

}

cout << i+1 << ". ";

cout << m[i].fio << "; "; if (m[i].d<10) cout << "0"; cout << m[i].d << "."; if (m[i].m<10) cout << "0"; cout << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

obaza << i+1 << ". ";

obaza << m[i].fio << "; "; if (m[i].d<10) obaza << "0"; obaza << m[i].d << "."; if (m[i].m<10) obaza << "0"; obaza << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

}

obaza.close();

mc=true;

}

void CUser(){

ofstream obaza("baza.txt");

cout << "Количество элементов: ";

cin >> r;

m = new Massiv[r];

obaza << "Количество элементов: " << r << endl;

cin.ignore(30,'\n');

for (int i=0;i<r;i++){

cout << i+1 << "." << endl << "ФИО: ";

getline (cin, m[i].fio);

cout << "День (число месяца) рождения: "; cin >> m[i].d;

cout << "Месяц рождения: " ; cin >> m[i].m;

cout << "Год рождения: "; cin >> m[i].g;

cout << "Адрес: "; cin.ignore(30,'\n'); getline (cin, m[i].adres);

cout << i+1 << ". ";

cout << m[i].fio << "; "; if (m[i].d<10) cout << "0"; cout << m[i].d << "."; if (m[i].m<10) cout << "0"; cout << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

obaza << i+1 << ". ";

obaza << m[i].fio << "; "; if (m[i].d<10) obaza << "0"; obaza << m[i].d << "."; if (m[i].m<10) obaza << "0"; obaza << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

}

obaza.close();

}

void Take(){

int i=0;

ifstream ibaza("baza.txt");

if (!ibaza) cout << "Файл не открыт. "; else cout << "Файл открыт успешно. "; cout << endl;

char sod[50],drop[30];

ibaza.getline(drop,30,':');

if (ibaza.eof()) cout << "Файл пустой. " << endl;

else{

ibaza >> r;

m = new Massiv [r];

for (int i=0;i<r;i++){

ibaza.getline(drop,5,' ');

ibaza.getline(sod,50,';');

m[i].fio = sod;

ibaza >> m[i].d;

ibaza.getline(drop,5,'.');

ibaza >> m[i].m;

ibaza.getline(drop,5,'.');

ibaza >> m[i].g;

ibaza.getline(drop,30,' ');

ibaza.getline(sod,50,';');

ibaza.getline(drop,30,'\n');

m[i].adres=sod;

cout << i+1 << ". " << m[i].fio << "; "; if (m[i].d<10) cout << "0"; cout << m[i].d << "."; if (m[i].m<10) cout << "0"; cout << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

}

ibaza.close();

cout << "Файл успешно закрыт. " << endl;

mc=true;

}

}

void Dop(){

ofstream obaza("baza.txt");

int nomer;

cout << "Номер, на позицию которого хотите добавить элемент: ";

cin >> nomer;

while (nomer<1||nomer>r+1||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать длину массива более чем на 1! Повторите ввод: ";

cin >> nomer;

}

r++;

obaza << "Количество элементов: " << r << endl;

Massiv \*nm = new Massiv [r];

for (int i=0;i<nomer-1;i++){

nm[i]=m[i];

cout << i+1 << ". " << nm[i].fio << "; "; if (nm[i].d<10) cout << "0"; cout << nm[i].d << "."; if (nm[i].m<10) cout << "0"; cout << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

obaza << i+1 << ". " << nm[i].fio << "; "; if (nm[i].d<10) obaza << "0"; obaza << nm[i].d << "."; if (nm[i].m<10) obaza << "0"; obaza << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

}

cout << "ФИО элемента: "; cin.ignore(30,'\n'); getline (cin, nm[nomer-1].fio);

cout << "День (число месяца), в который родился элемент: "; cin >> nm[nomer-1].d;

while (nm[nomer-1].d<1||nm[nomer-1].d>31||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать 31! Повторите ввод: ";

cin >> nm[nomer-1].d;

}

cout << "Месяц рождения элемента: ";

cin >> nm[nomer-1].m;

while (nm[nomer-1].m<1||nm[nomer-1].m>12||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать 12! Повторите ввод: ";

cin >> nm[nomer-1].m;

}

cout << "Год рождения элемента: ";

cin >> nm[nomer-1].g;

while (nm[nomer-1].g<1900||nm[nomer-1].g>2021||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть больше 1900 и меньше 2021! Повторите ввод: ";

cin >> nm[nomer-1].g;

}

cout << "Адрес элемента: "; cin.ignore(30,'\n'); getline (cin, nm[nomer-1].adres);

cout << nomer << ". " << nm[nomer-1].fio << "; "; if (nm[nomer-1].d<10) cout << "0"; cout << nm[nomer-1].d << "."; if (nm[nomer-1].m<10) cout << "0"; cout << nm[nomer-1].m << "." << nm[nomer-1].g << "; " << nm[nomer-1].adres << ";" << endl;

obaza << nomer << ". " << nm[nomer-1].fio << "; "; if (nm[nomer-1].d<10) obaza << "0"; obaza << nm[nomer-1].d << "."; if (nm[nomer-1].m<10) obaza << "0"; obaza << nm[nomer-1].m << "." << nm[nomer-1].g << "; " << nm[nomer-1].adres << ";" << endl;

for (int i=nomer;i<r;i++){

nm[i]=m[i-1];

cout << i+1 << ". " << nm[i].fio << "; "; if (nm[i].d<10) cout << "0"; cout << nm[i].d << "."; if (nm[i].m<10) cout << "0"; cout << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

obaza << i+1 << ". " << nm[i].fio << "; "; if (nm[i].d<10) obaza << "0"; obaza << nm[i].d << "."; if (nm[i].m<10) obaza << "0"; obaza << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

}

cout << "Массив обновлён. " << endl;

delete [] nm;

Take();

obaza.close();

}

void Udalit(){

ofstream obaza("baza.txt");

int nomer;

cout << "Номер удаляемого элемента: ";

cin >> nomer;

while (nomer<1||nomer>r||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать длину массива! Повторите ввод: ";

cin >> nomer;

}

r--;

obaza << "Количество элементов: " << r << endl;

Massiv \*nm = new Massiv [r];

for (int i=0;i<nomer-1;i++){

nm[i]=m[i];

obaza << i+1 << ". " << nm[i].fio << "; "; if (nm[i].d<10) obaza << "0"; obaza << nm[i].d << "."; if (nm[i].m<10) obaza << "0"; obaza << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

}

for (int i=nomer;i<r+1;i++){

nm[i-1]=m[i];

obaza << i << ". " << nm[i].fio << "; "; if (nm[i].d<10) obaza << "0"; obaza << nm[i].d << "."; if (nm[i].m<10) obaza << "0"; obaza << nm[i].m << "." << nm[i].g << "; " << nm[i].adres << ";" << endl;

}

cout << "Массив обновлён. " << endl;

delete [] nm;

Take();

}

void Set(){

const int fon = system("Color F0");

system("chcp 1251>nul");

}

void LineMetod(){

string key;

bool nai=false;

cout << "Запущен линейный метод поиска. " << endl;

cout << "Адрес человека, которого необходимо найти: "; getline(cin, key);getline(cin, key);

for (int i=0;i<r;i++){

if (key==m[i].adres){

cout << "Элемент номер " << i+1 << ": " << m[i].fio << "; "; if (m[i].d<10) cout << "0"; cout << m[i].d << "."; if (m[i].m<10) cout << "0"; cout << m[i].m << "." << m[i].g << "; " << m[i].adres << endl;

nai=true;

}

}

if (nai==false) cout << "Элемент не найден. " << endl;

}

void SortiAdres(){

ofstream obaza("baza.txt");

obaza << "Количество элементов: " << r << endl;

int nom;

string adr;

Massiv k;

for (int i=0;i<r;i++){

adr=m[i].adres;

k=m[i];

nom=i;

for (int e=i;e<r;e++) if (m[e].adres<adr) {adr=m[e].adres;k = m[e];nom=e;}

m[nom]=m[i];

m[i]=k;

}

cout << "Массив отсортирован. " << endl;

for (int i=0;i<r;i++){

cout << i+1 << ". ";

cout << m[i].fio << "; "; if (m[i].d<10) cout << "0"; cout << m[i].d << "."; if (m[i].m<10) cout << "0"; cout << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

obaza << i+1 << ". ";

obaza << m[i].fio << "; "; if (m[i].d<10) obaza << "0"; obaza << m[i].d << "."; if (m[i].m<10) obaza << "0"; obaza << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

}

obaza.close();

}

void SortiData(){

ofstream obaza("baza.txt");

obaza << "Количество элементов: " << r << endl;

int nom,sum,cur;

Massiv k;

for (int i=0;i<r;i++){

sum=m[i].g\*10000+m[i].m\*100+m[i].d;

nom=i;

k=m[i];

for (int e=i;e<r;e++){

cur = m[e].g\*10000+m[e].m\*100+m[e].d;

if (cur<sum) {sum=cur;k=m[e];nom=e;}

}

m[nom]=m[i];

m[i]=k;

}

cout << "Массив отсортирован. " << endl;

for (int i=0;i<r;i++){

cout << i+1 << ". ";

cout << m[i].fio << "; "; if (m[i].d<10) cout << "0"; cout << m[i].d << "."; if (m[i].m<10) cout << "0"; cout << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

obaza << i+1 << ". ";

obaza << m[i].fio << "; "; if (m[i].d<10) obaza << "0"; obaza << m[i].d << "."; if (m[i].m<10) obaza << "0"; obaza << m[i].m << "." << m[i].g << "; " << m[i].adres << ";" << endl;

}

obaza.close();

}

void Interpolar(){

SortiData();

string key;

int middle, left=0, right=r-1, ztchk=0,curdat=0;

bool nai=false;

cout << "Запупщен интерполяционный метод поиска. " << endl;

cout << "Дата рождения человека, которого необходимо найти: " << endl << "День: "; cin >> ztchk;

while (ztchk<1||ztchk>31||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать 31! Повторите ввод: ";

cin >> ztchk;

}

curdat=ztchk;

cout << "Месяц рождения элемента: ";

cin >> ztchk;

while (ztchk<1||ztchk>12||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть положительным и не должно превышать 12! Повторите ввод: ";

cin >> ztchk;

}

curdat+=ztchk\*100;

cout << "Год рождения элемента: ";

cin >> ztchk;

while (ztchk<1900||ztchk>2021||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Введённое значение должно быть больше 1900 и меньше 2021! Повторите ввод: ";

cin >> ztchk;

}

curdat+=ztchk\*10000;

while (!nai&&left!=right){

middle=left+((curdat-(m[left].d+m[left].m\*100+m[left].g\*10000))\*(right-left))/((m[right].d+m[right].m\*100+m[right].g\*10000)-(m[left].d+m[left].m\*100+m[left].g\*10000));

if (m[middle].d+m[middle].m\*100+m[middle].g\*10000==curdat) {

cout << "Элемент номер " << middle+1 << ": " << m[middle].fio << "; "; if (m[middle].d<10) cout << "0"; cout << m[middle].d << "."; if (m[middle].m<10) cout << "0"; cout << m[middle].m << "." << m[middle].g << "; " << m[middle].adres << ";" << endl;

nai=true;

}

else{

if (m[middle].d+m[middle].m\*100+m[middle].g\*10000<curdat) left=middle+1;

else right = middle-1;

}

}

if (!nai) cout << "Элемент не найден. " << endl;

}

void Rovniy(){

string kucha,getted;

int e=0,nomber=0,dlin=0,sovpalo;

bool nai=false;

for (int i=0;i<r;i++) {kucha+="\n";kucha+=m[i].adres;}

cout << "Адрес искомого элемента: ";

cin.ignore(30,'\n');

getline(cin,getted);

for (int i=0;i<kucha.length();i++){

dlin++;

if (kucha[i]=='\n') nomber++;

if (getted[e++]==kucha[i]) {

sovpalo++;

}

else {e=0;sovpalo=0;}

if (sovpalo==getted.length()&&sovpalo==dlin){

nai=true;

sovpalo=0;

e=0;

cout << "Элемент, содержащий введённые данные: " << nomber << ". " << m[nomber-1].fio << "; "; if (m[nomber-1].d<10) cout << "0"; cout << m[nomber-1].d << "."; if (m[nomber-1].m<10) cout << "0"; cout << m[nomber-1].m << "." << m[nomber-1].g << "; " << m[nomber-1].adres << endl;

}

if (kucha[i]=='\n') dlin=0;

}

if (!nai) cout << "Элемент не найден. " << endl;

}

int main(){

Set();

while (v!=0){

cout << "Выберете действие: " << endl << "0. Завершить работу программы. \n1. Сформировать массив случайно; \n2. Сформировать массив самостоятельно; \n3. Скопировать массив из файла; \n4. Удалить элемент из массива; \n5. Добавить элемент в массив; ";

cout <<"\n6. Запустить линейный метод поиска; \n7. Запустить интерполяционный метод поиска; \n8. Запустить прямой метод поиска. " << endl;

cin >> v;

switch(v){

case 0: return 0;

case 1: Create();break;

case 2: CUser(); break;

case 3: Take(); break;

case 4: if (mc==true) Udalit(); else cout << "База данных отсутствует. " << endl; break;

case 5: if (mc==true) Dop(); else cout << "База данных отсутствует. " << endl; break;

case 6: if (mc==true) LineMetod(); else cout << "База данных отсутствует. " << endl; break;

case 7: if (mc==true) Interpolar(); else cout << "База данных отсутствует. " << endl; break;

case 8: if (mc==true) Rovniy(); else cout << "База данных отсутствует. " << endl; break;

default: cout << "Неверный ввод! " << endl;

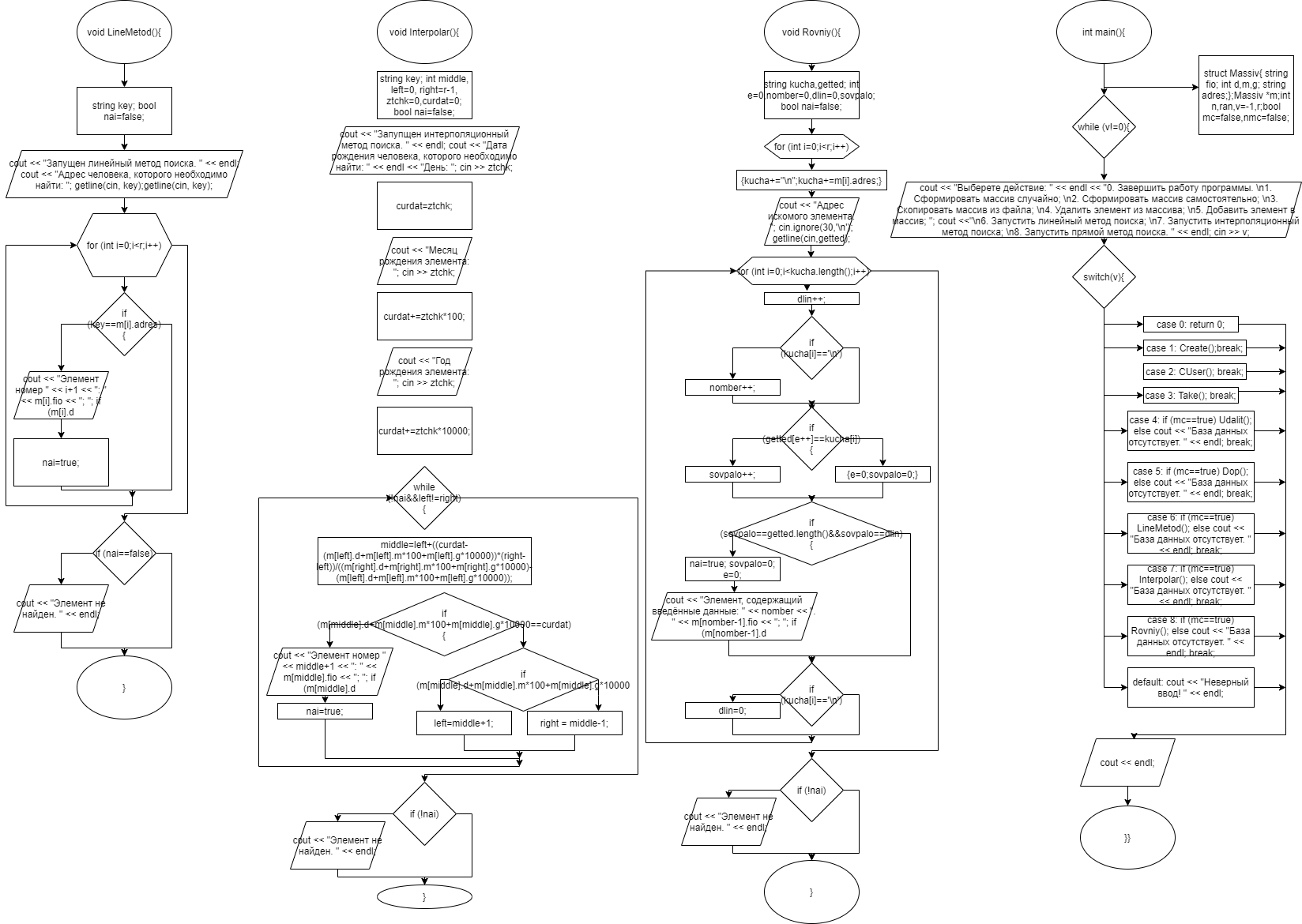
}

cout << endl;

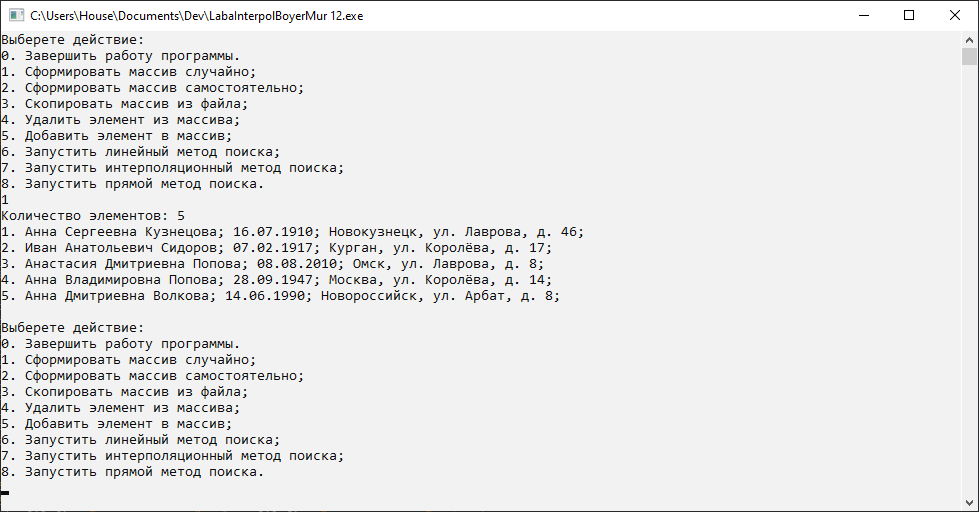
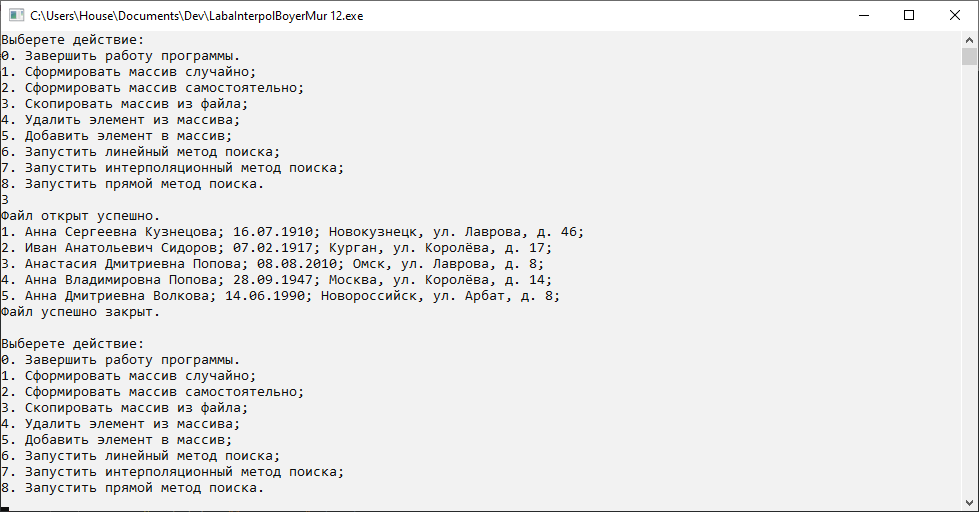
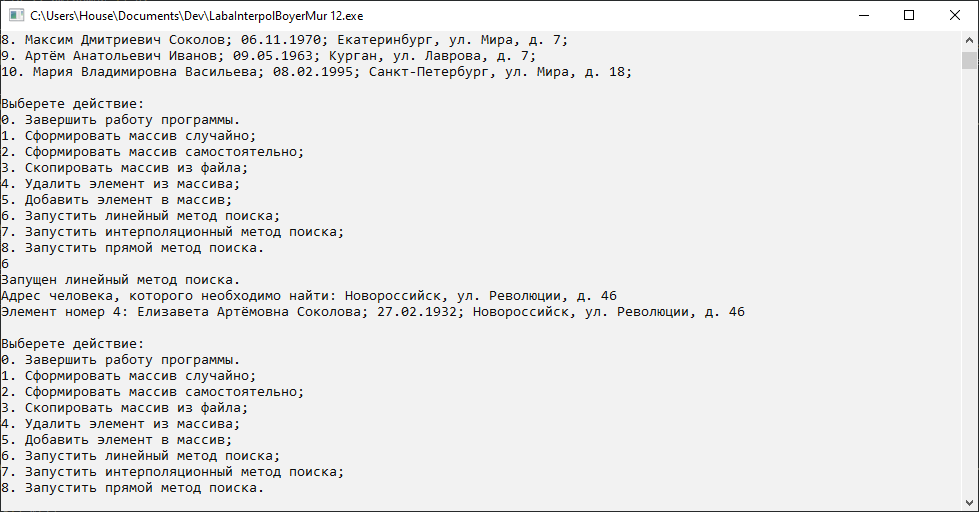
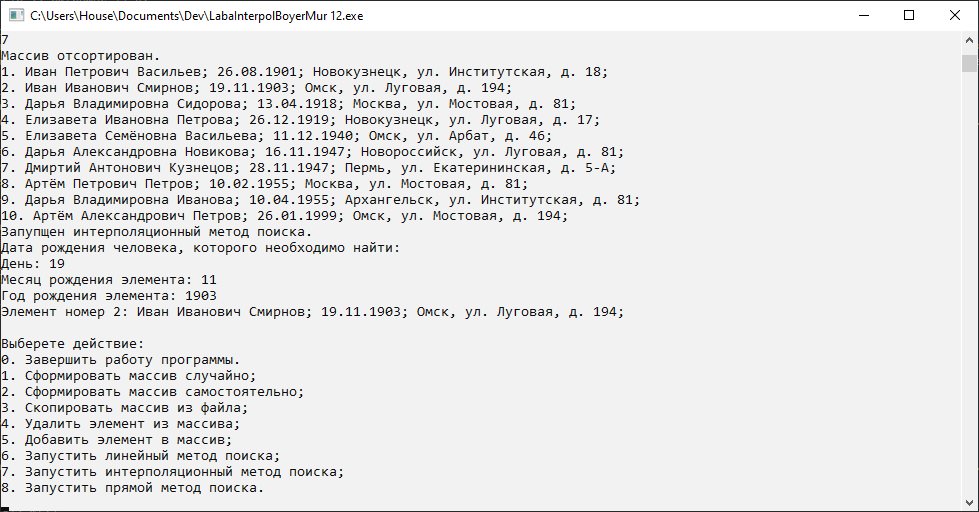
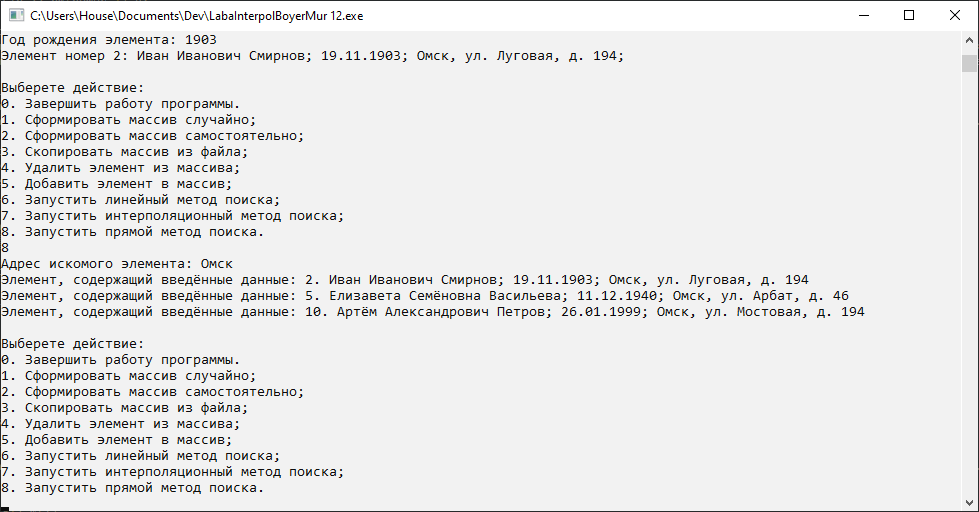
}

}

**Блок-схема**



**Результаты тестов**

**    **