Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №17**

Дисциплина: основы теории алгоритмов и структуры данных

Тема: Метод быстрого поиска. Хеширование

Вариант: 25

Выполнил работу

студент группы РИС-20-1б

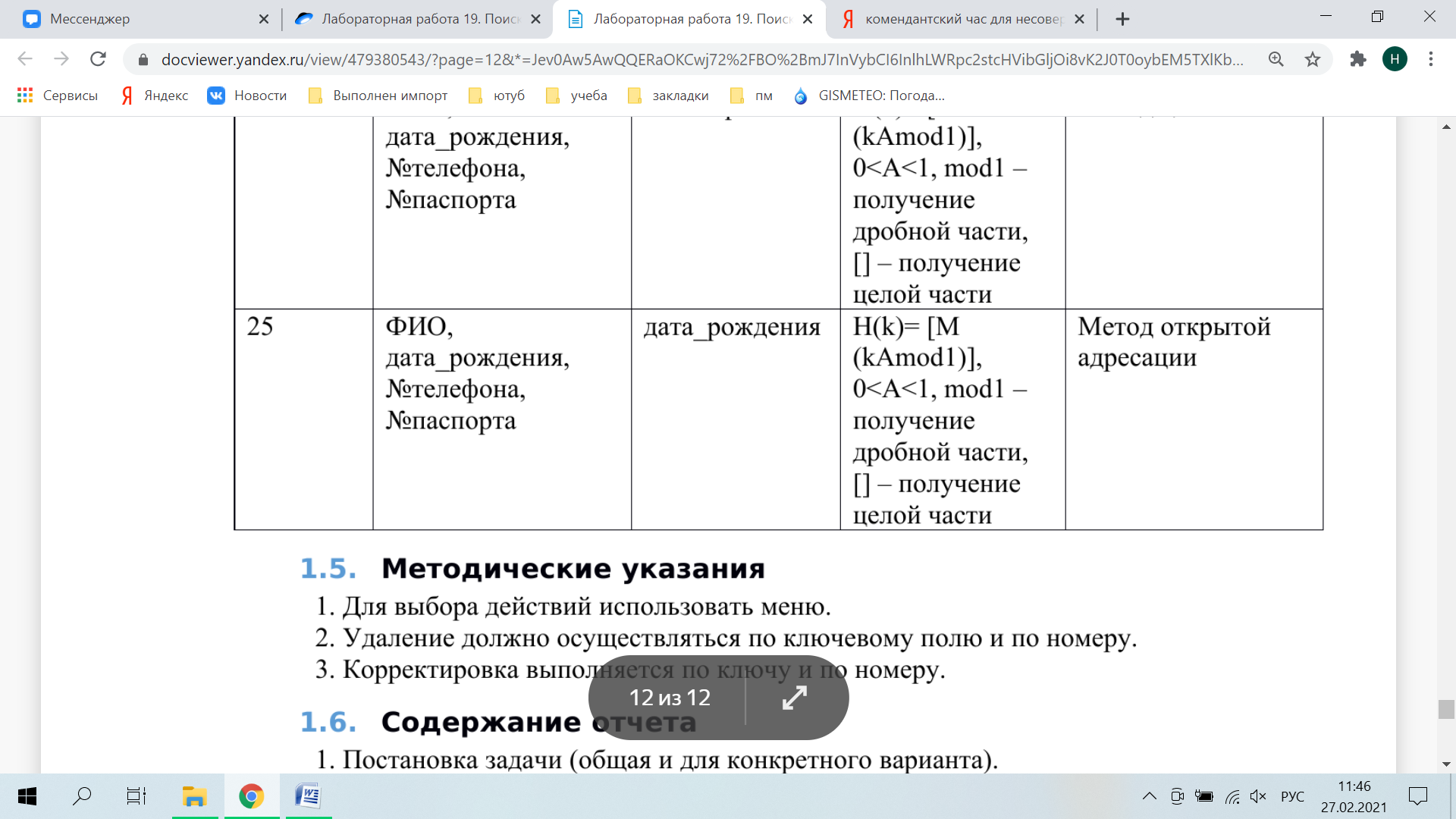
Рябов Никита Андреевич

Проверила

Доцент кафедры ИТАС Полякова О.А.

Пермь, 2021

**Постановка задачи**



**Анализ задачи**

*Какие предстоит выполнить действия.*

Считывание размера таблицы, заполнение «пустыми» элементами.

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int len=-1;

int adress;

CITIZEN x;

string user\_search;

int error=0;

while (len<2 || len>100)

{

cout << "Введите размер таблицы\n";

cin >> len;

}

const int N=len;

cin.get();

CITIZEN \*data\_base=new CITIZEN[len];

string \*key=new string[len];

for (int i=0;i<len;i++) key[i]="404";

Считывание данных из файла, получение результата от хеш-функции. При необходимости увеличение таблицы. Запись таблицу.

for (int i=0;i<N;i++)

{

getline(F1,x.fio);

getline(F1,x.birthday);

getline(F1,x.phone);

getline(F1,x.passport);

adress=Hash(x.birthday,N);

while (adress>=len || key[len-1]!="404")

{

CITIZEN \*data\_base\_new=new CITIZEN[len\*2];

string \*key\_new=new string[len\*2];

for (int i=0;i<len;i++)

{

data\_base\_new[i]=data\_base[i];

key\_new[i]=key[i];

}

for (int i=len;i<len\*2;i++) key\_new[i]="404";

delete [] data\_base;

delete [] key;

data\_base=data\_base\_new;

key=key\_new;

len\*=2;

}

if (key[adress]!="404")

{

error++;

while (key[adress]!="404") adress++;

}

data\_base[adress]=x;

key[adress]=x.birthday;

}

Вывод хеш-таблицы

for (int i=0; i<len;i++)

{

cout << i << endl;

cout << key[i] << endl;

cout << "ФИО - " << data\_base[i].fio << endl;

cout << "Дата рождения - " << data\_base[i].birthday << endl;

cout << "Номер телефона - " << data\_base[i].phone << endl;

cout << "Номер паспорта - " << data\_base[i].passport << endl;

}

Получение запроса пользователя. Линейный поиск нужного элемента, начиная с номера, выданного хеш-функцией. Поиск до нахождения элемента или до нахождения пустой ячейки

cout << "Введите запрос\n";

cin >> user\_search;

adress=Hash(user\_search,N);

while (adress<len)

{

if (key[adress]==user\_search)

{

cout << "Объект найден:\n";

cout << "ФИО - " << data\_base[adress].fio << endl;

cout << "Дата рождения - " << data\_base[adress].birthday << endl;

cout << "Номер телефона - " << data\_base[adress].phone << endl;

cout << "Номер паспорта - " << data\_base[adress].passport << endl;

break;

}

if (key[adress]=="404")

{

cout << "Неизвестный объект [404]\n";

break;

}

adress++;

}

if (adress==len) cout << "Неизвестный объект [404]\n";

Хеш-функция

int Hash (string s,int n)

{

int x=0;

for (int i=0;i<s.size();i++) x+=(int)(s[i])\*(i+1);

return (int)(n\*(0.3\*x-(int)(0.3\*x)));

}

*С каким типом данных действуем*

Строки

*В каком виде представлены данные*

Массив структур

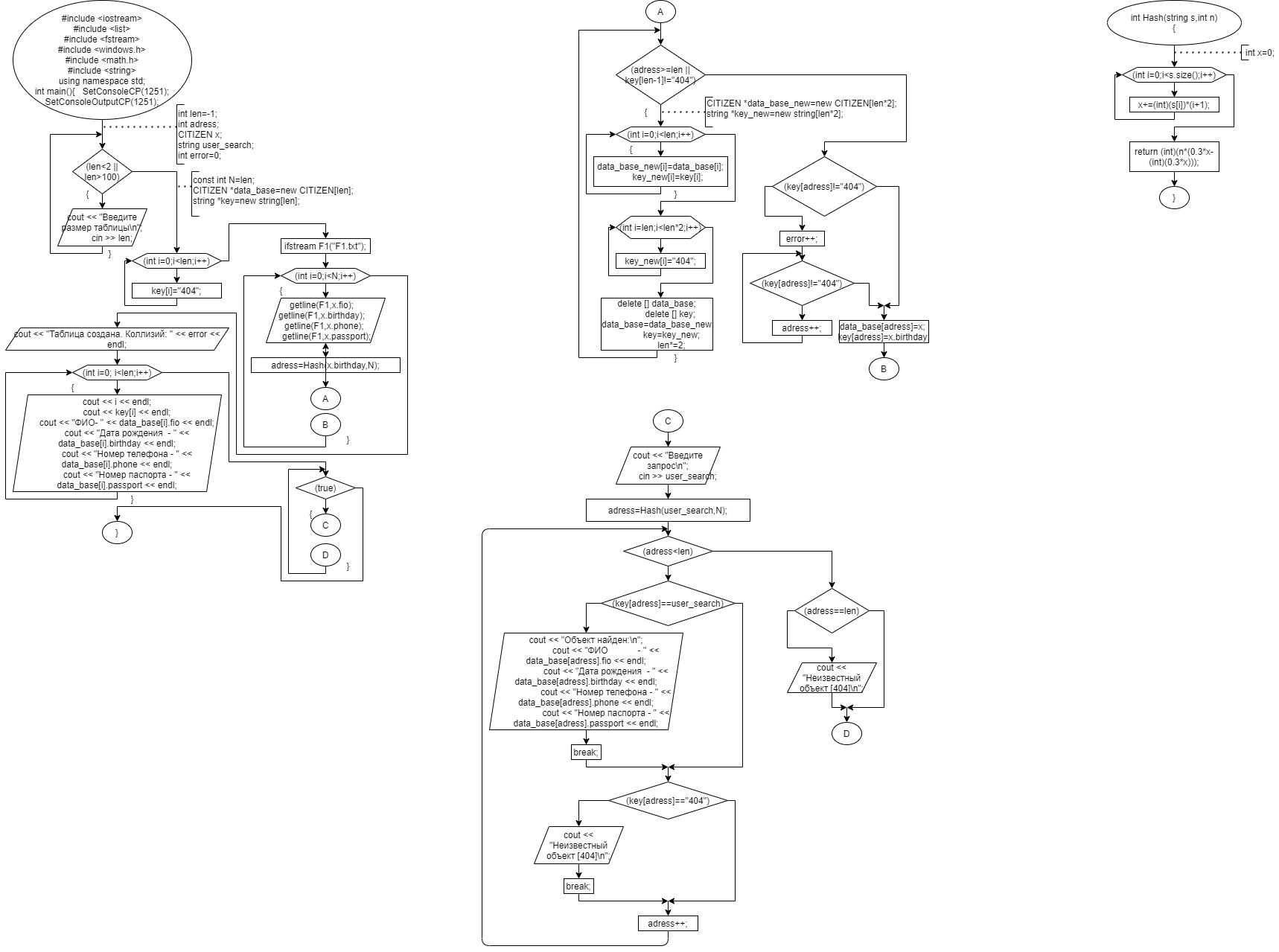
*Какие поля*

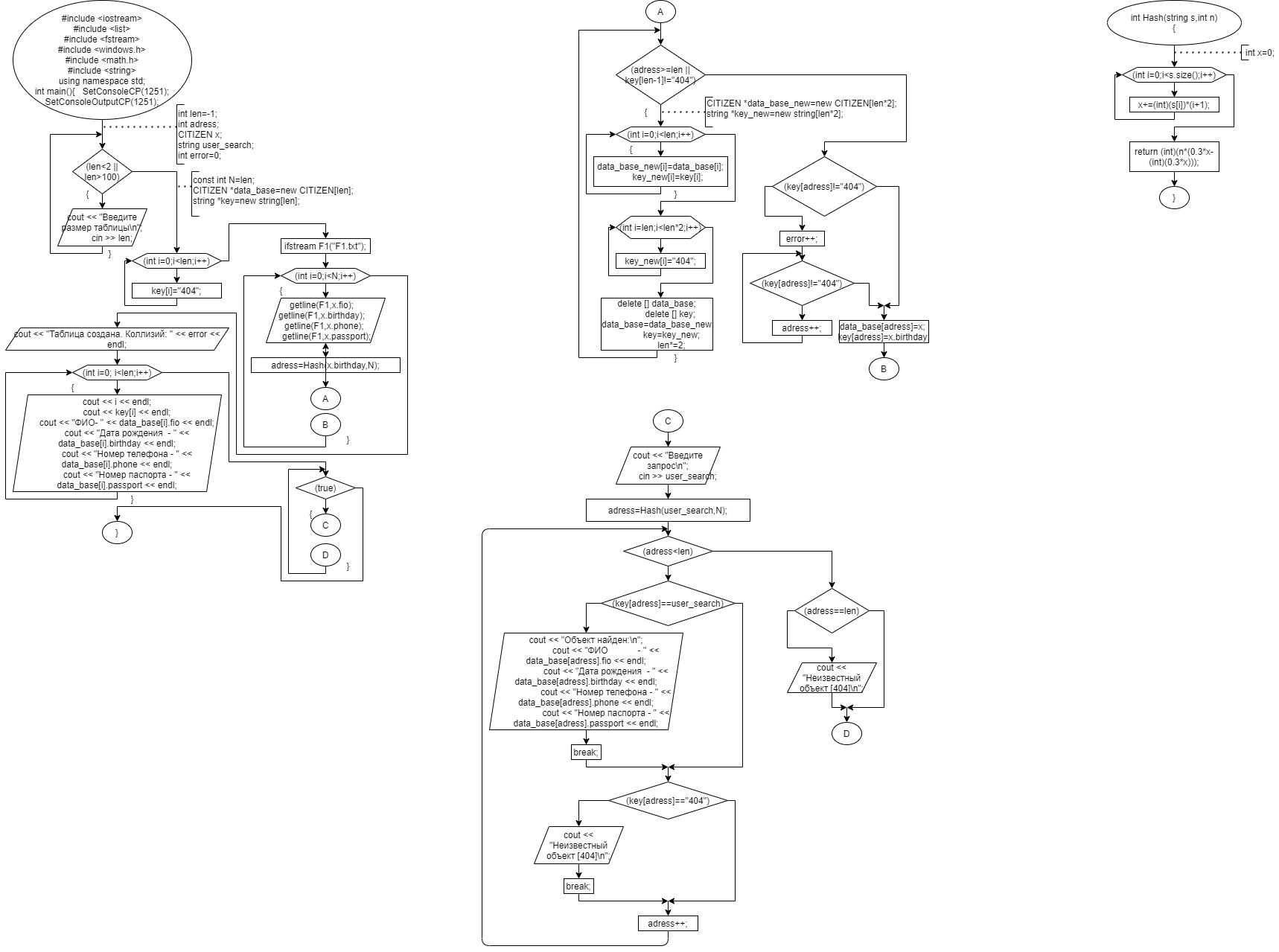
4 строковых поля: ФИО, дата рождения, номер телефона, номер паспорта

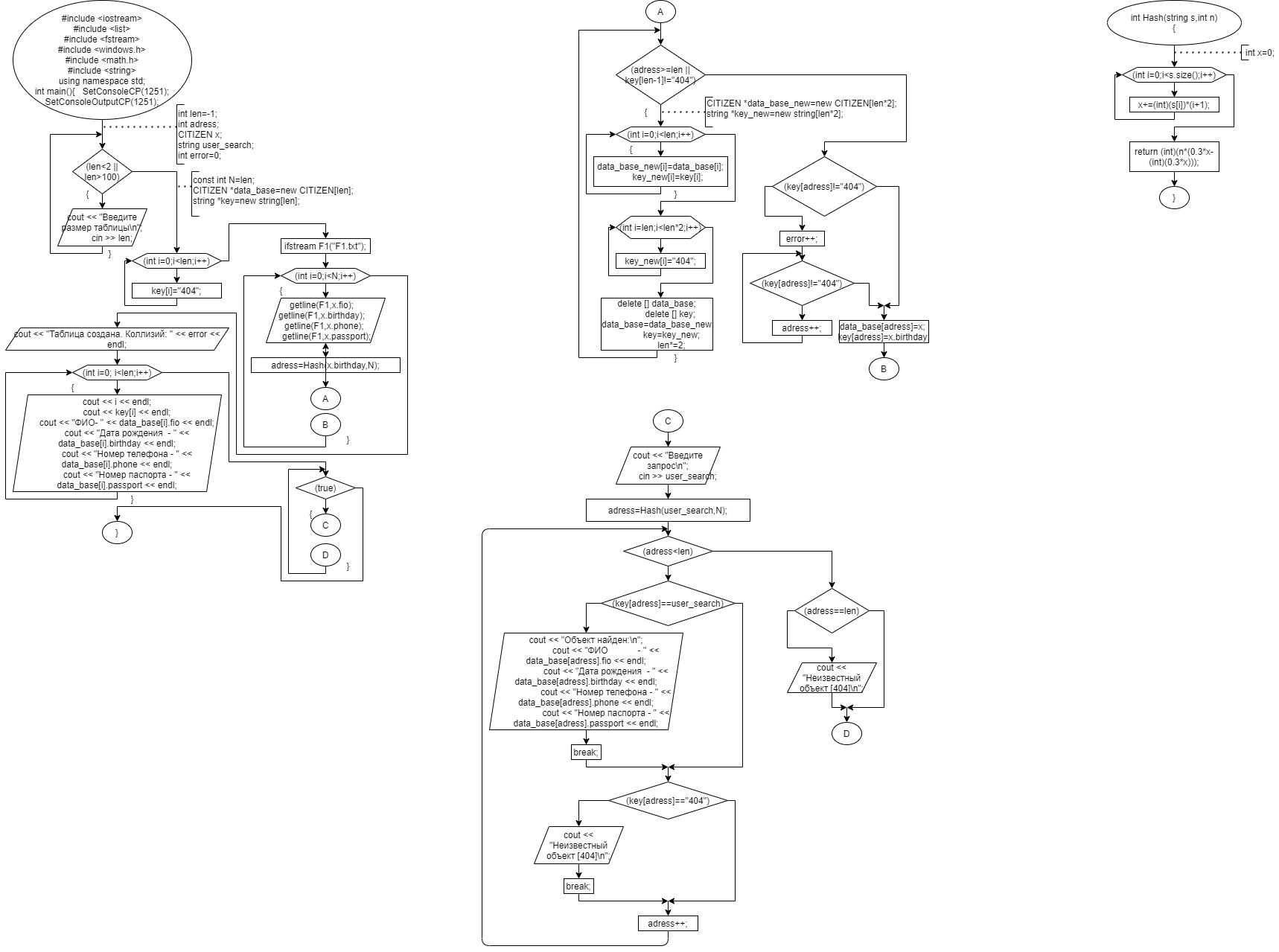
*Операторы ввода/вывода*

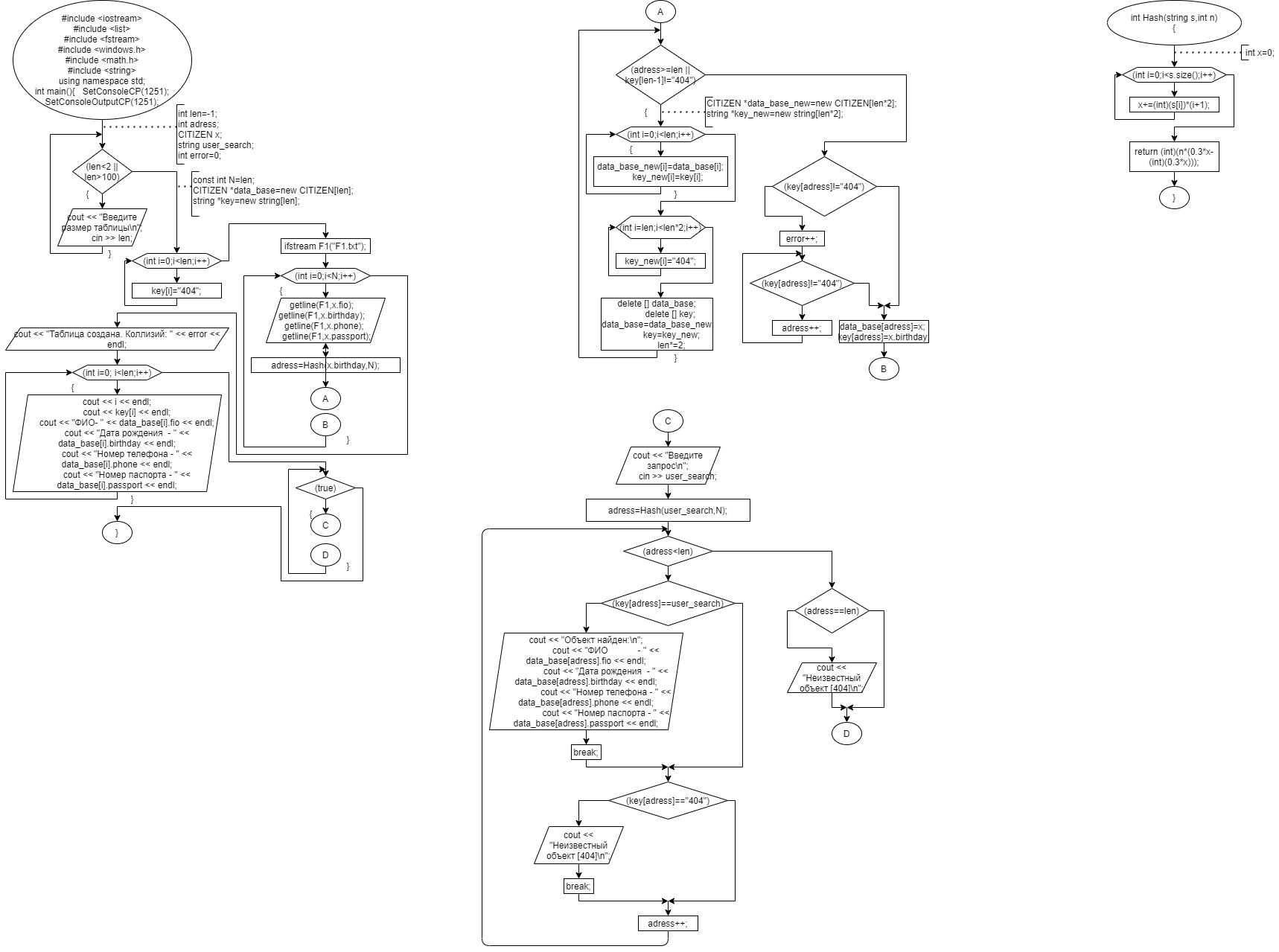
В программе используются операторы cin/cout. Getline для считывания из файла

**Блок-схема**

****

****

****

****

**Код**

#include <iostream>

#include <list>

#include <fstream>

#include <windows.h>

#include <math.h>

#include <string>

using namespace std;

struct CITIZEN

{

string fio;

string birthday;

string phone;

string passport;

};

int Hash\_task(string s,int n)

{

int x=0;

for (int i=0;i<s.size();i++) x+=(int)(s[i])\*(i+1);

return (int)(n\*(0.3\*x-(int)(0.3\*x)));

}

int Hash\_quadro(string s,int n)

{

int x=0;

int l;

for (int i=0;i<s.size();i++) x+=(int)(s[i])\*(i+1);

x=x\*x;

l=to\_string(x).size();

return stoi(to\_string(x).substr(l/3,l\*2/3));

}

int Hash\_ss(string s,int n)

{

int x=0;

int result;

for (int i=0;i<s.size();i++) x+=(int)(s[i])\*(i+1);

for (int i=0;i<to\_string(x).size();i++) result+=x/((int)pow(10,i))%10\*((int)pow(11,i));

return result/10;

}

int SwitchHash(string s, int n)

{

int x=1;

switch (x)

{

case 1:

return Hash\_task(s, n);

case 2:

return Hash\_quadro(s, n);

case 3:

return Hash\_ss(s, n);

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int len=-1;

int adress;

CITIZEN x;

string user\_search;

int error=0;

while (len<2 || len>100)

{

cout << "Введите размер таблицы\n";

cin >> len;

}

const int N=len;

cin.get();

CITIZEN \*data\_base=new CITIZEN[len];

string \*key=new string[len];

for (int i=0;i<len;i++) key[i]="404";

ifstream F1("F1.txt");

for (int i=0;i<N;i++)

{

getline(F1,x.fio);

getline(F1,x.birthday);

getline(F1,x.phone);

getline(F1,x.passport);

adress=SwitchHash(x.birthday,N);

while (adress>=len || key[len-1]!="404")

{

CITIZEN \*data\_base\_new=new CITIZEN[len\*2];

string \*key\_new=new string[len\*2];

for (int i=0;i<len;i++)

{

data\_base\_new[i]=data\_base[i];

key\_new[i]=key[i];

}

for (int i=len;i<len\*2;i++) key\_new[i]="404";

delete [] data\_base;

delete [] key;

data\_base=data\_base\_new;

key=key\_new;

len\*=2;

}

if (key[adress]!="404")

{

error++;

while (key[adress]!="404") adress++;

}

data\_base[adress]=x;

key[adress]=x.birthday;

}

cout << "Таблица создана. Коллизий: " << error << endl;

for (int i=0; i<len;i++)

{

cout << i << endl;

cout << key[i] << endl;

cout << "ФИО - " << data\_base[i].fio << endl;

cout << "Дата рождения - " << data\_base[i].birthday << endl;

cout << "Номер телефона - " << data\_base[i].phone << endl;

cout << "Номер паспорта - " << data\_base[i].passport << endl;

}

while (true)

{

cout << "Введите запрос\n";

cin >> user\_search;

adress=SwitchHash(user\_search,N);

while (adress<len)

{

if (key[adress]==user\_search)

{

cout << "Объект найден:\n";

cout << "ФИО - " << data\_base[adress].fio << endl;

cout << "Дата рождения - " << data\_base[adress].birthday << endl;

cout << "Номер телефона - " << data\_base[adress].phone << endl;

cout << "Номер паспорта - " << data\_base[adress].passport << endl;

break;

}

if (key[adress]=="404")

{

cout << "Неизвестный объект [404]\n";

break;

}

adress++;

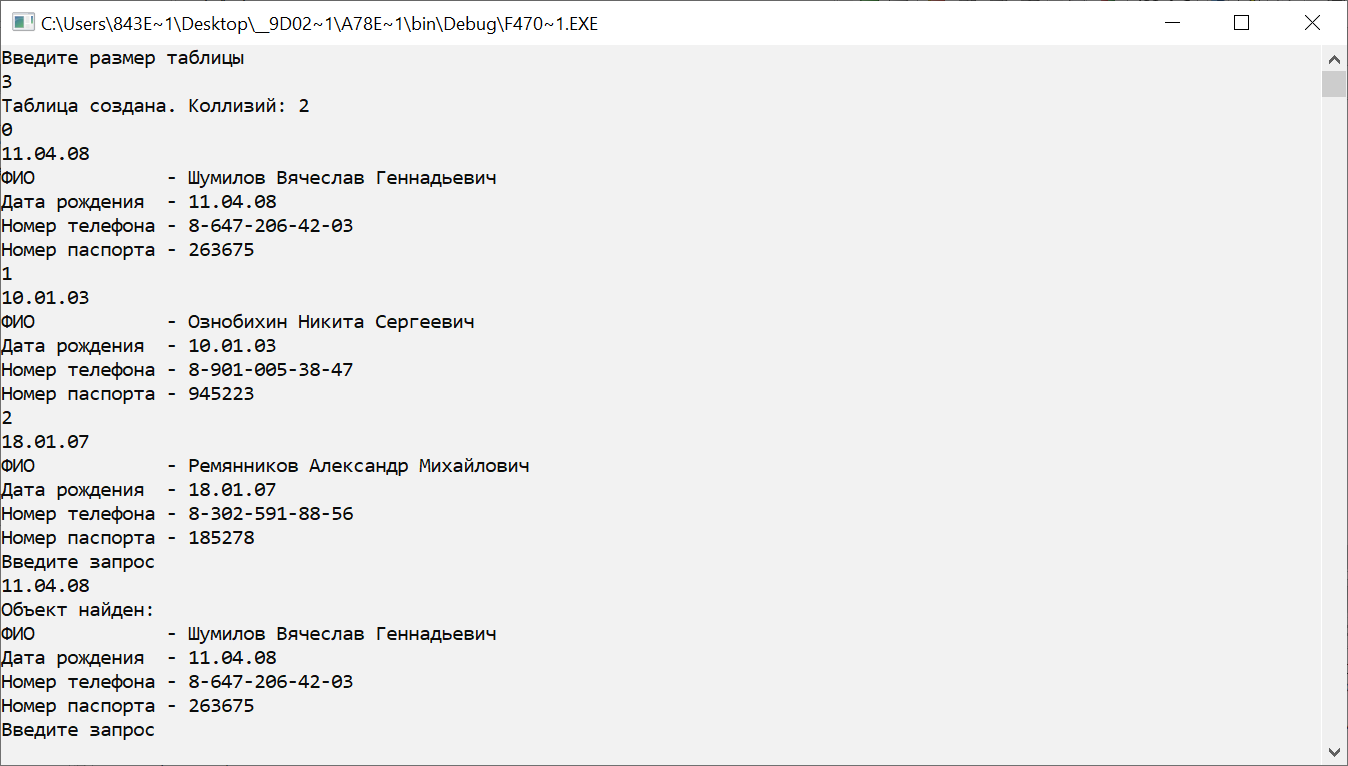
}

if (adress==len) cout << "Неизвестный объект [404]\n";

}

}

**Скриншоты**

****