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 №17**

Дисциплина: «Информатика»

Тема: Методы быстрого поиска. Хеширование

Вариант 4

Выполнил:

Студент группы РИС-20-1б

Еске Вячеслав Сергеевич

Проверила:

Доцент кафедры ИТАС

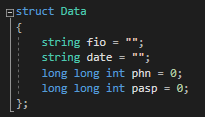
Полякова О. А.

**Постановка задачи**

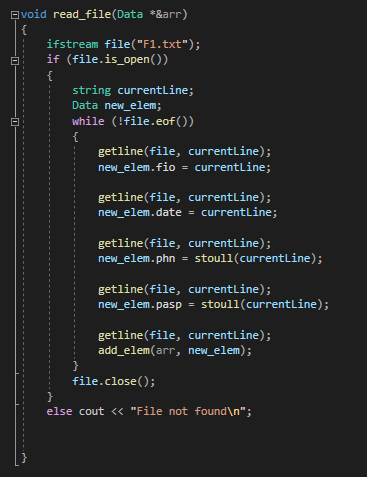
1. В качестве данных используются ФИО, №паспорта, адрес. С помощью ключа поиска, которым в данном варианте является №паспорта, и с помощью хэш—функции H(k)= [M(kAmod1)],0<A<1, mod1 –получение дробной части,[] – получение целой части, а также в качестве метода рехеширования используя метод цепочек, написать программу, которая будет выполнять поиск по заданным изначально данным (данные находятся в текстовых документах).

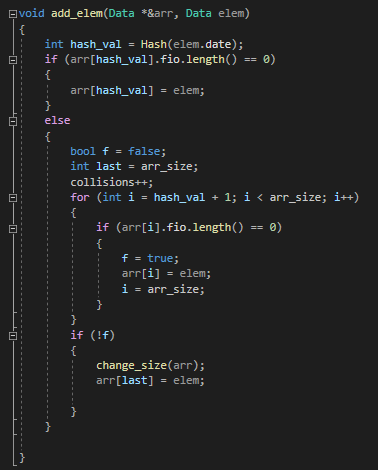
**Анализ задачи**

1. Для решения задачи необходимо…
   1. Создать структуру Data для хранения данных различных типов в программе для удобства обращения к ним;

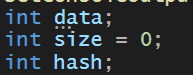


* 1. Считывание данных из файла и добавление элементов в хеш-таблицу;





1. В программе были использованы следующие типы данных:
   1. Тип данных int для хранения различных промежуточных данных в программе;



* 1. Тип данных string для хранения различной информации о человеке, например, его ФИО;



**Код**

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

int arr\_size, collisions, M;

template <typename T>

void get\_value(T& value)

{

bool is\_correct = false;

do

{

cout << "Введите " << "элемент\n";

cin >> value;

if (cin.fail())

{

cout << "\nНекорректное значение" << endl << endl;

cin.clear();

cin.ignore(32767, '\n');

}

else

{

is\_correct = true;

cin.ignore(32767, '\n');

}

} while (!is\_correct);

}

struct Data

{

string fio = "";

string date = "";

long long int phn = 0;

string pasp = "";

};

int Hash(string date)

{

return (stoi(date.substr(0, 2)) + stoi(date.substr(3, 5)) + stoi(date.substr(6, 10))) % M;

}

void change\_size(Data \*&arr)

{

Data\* new\_arr = new Data[arr\_size + (arr\_size / 2)];

for (int i = 0; i < arr\_size; i++)

{

new\_arr[i] = arr[i];

}

arr\_size += arr\_size / 2;

delete[] arr;

arr = new\_arr;

}

void add\_elem(Data \*&arr, Data elem)

{

int hash\_val = Hash(elem.date);

if (arr[hash\_val].fio.length() == 0)

{

arr[hash\_val] = elem;

}

else

{

bool f = false;

int last = arr\_size;

collisions++;

for (int i = hash\_val + 1; i < arr\_size; i++)

{

if (arr[i].fio.length() == 0)

{

f = true;

arr[i] = elem;

i = arr\_size;

}

}

if (!f)

{

change\_size(arr);

arr[last] = elem;

}

}

}

void read\_file(Data \*&arr)

{

ifstream file("F1.txt");

if (file.is\_open())

{

string currentLine;

Data new\_elem;

while (!file.eof())

{

getline(file, currentLine);

new\_elem.fio = currentLine;

getline(file, currentLine);

new\_elem.date = currentLine;

getline(file, currentLine);

new\_elem.phn = stoull(currentLine);

getline(file, currentLine);

new\_elem.pasp = currentLine;

getline(file, currentLine);

add\_elem(arr, new\_elem);

}

file.close();

}

else cout << "File not found\n";

}

void print\_hash\_table(Data arr[])

{

for (int i = 0; i < arr\_size; i++)

{

if (arr[i].fio.length() != 0) {

cout << "\nНомер элемента " << i << endl;

cout << "ФИО: " << arr[i].fio << endl;

cout << "Дата: " << arr[i].date << endl;

cout << "Номер телефона: " << arr[i].phn << endl;

cout << "Паспорта: " << arr[i].pasp << endl;

cout << "Значение хэша: " << Hash(arr[i].date) << endl << endl;

}

else

{

cout << "Элемент № " << i << " пуст" << endl;

}

}

}

bool is\_date(string date)

{

if (date.length() == 10)

{

return true;

}

else return false;

}

void find(Data arr[], string key)

{

if (is\_date(key))

{

int hash\_val = Hash(key);

if (arr[hash\_val].pasp == key)

{

cout << "Объект найден" << endl;

cout << "ФИО: " << arr[hash\_val].fio << endl;

cout << "Дата: " << arr[hash\_val].date << endl;

cout << "Номер телефона: " << arr[hash\_val].phn << endl;

cout << "Паспорта: " << arr[hash\_val].pasp << endl;

cout << "Значение хэша: " << hash\_val << endl << endl;

}

else

{

bool f = false;

for (int i = hash\_val + 1; i < arr\_size; i++)

{

if (arr[i].pasp == key)

{

f = true;

cout << "Объект найден" << endl;

cout << "ФИО: " << arr[hash\_val].fio << endl;

cout << "Дата: " << arr[hash\_val].date << endl;

cout << "Номер телефона: " << arr[hash\_val].phn << endl;

cout << "Паспорта: " << arr[hash\_val].pasp << endl;

cout << "Значение хэша: " << hash\_val << endl << endl;

}

}

if (!f)

{

cout << "Объект не найден\n";

}

}

}

else cout << "Некорректное значение\n";

}

int main()

{

system("chcp 1251>null");

do

{

get\_value(arr\_size);

} while (arr\_size < 2 || arr\_size > 100);

M = arr\_size;

Data\* data\_base = new Data[arr\_size];

read\_file(data\_base);

print\_hash\_table(data\_base);

cout << "Колличество коллизий: " << collisions << endl;

string key;

while (key != "0")

{

cout << "\nВведите искомый номер поспорта: ";

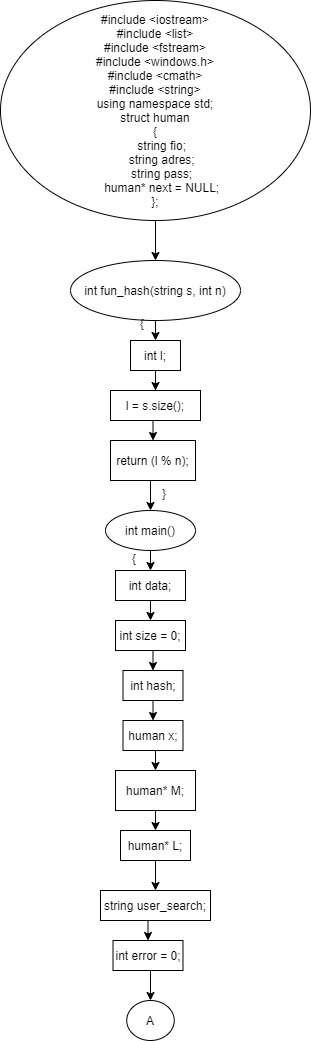
getline(cin, key);

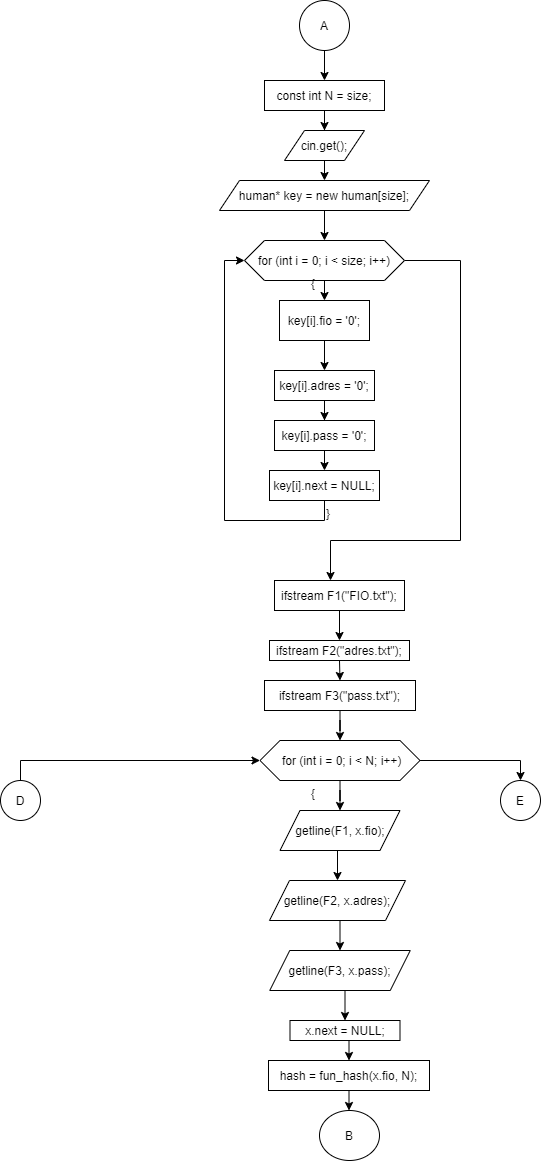
if (key != "0") find(data\_base, key);

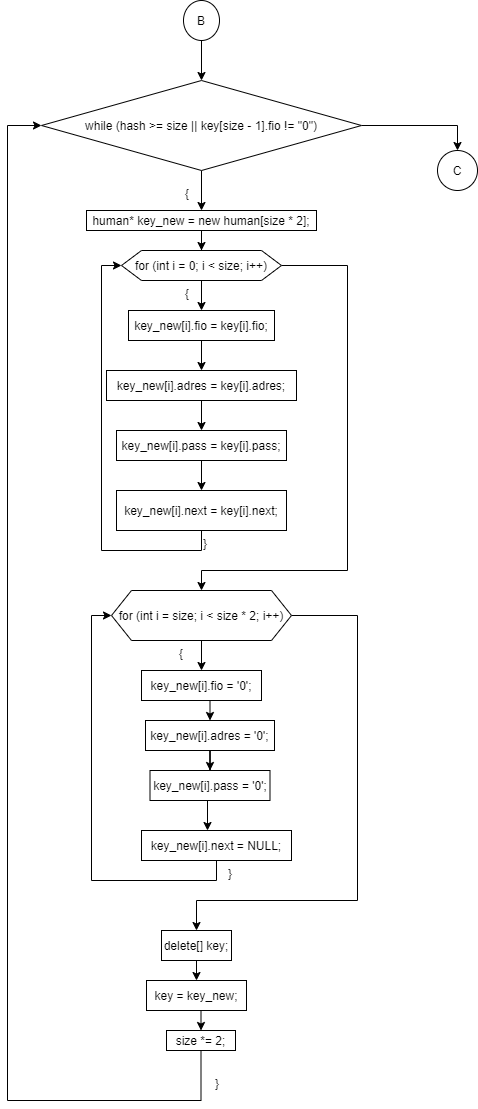
}

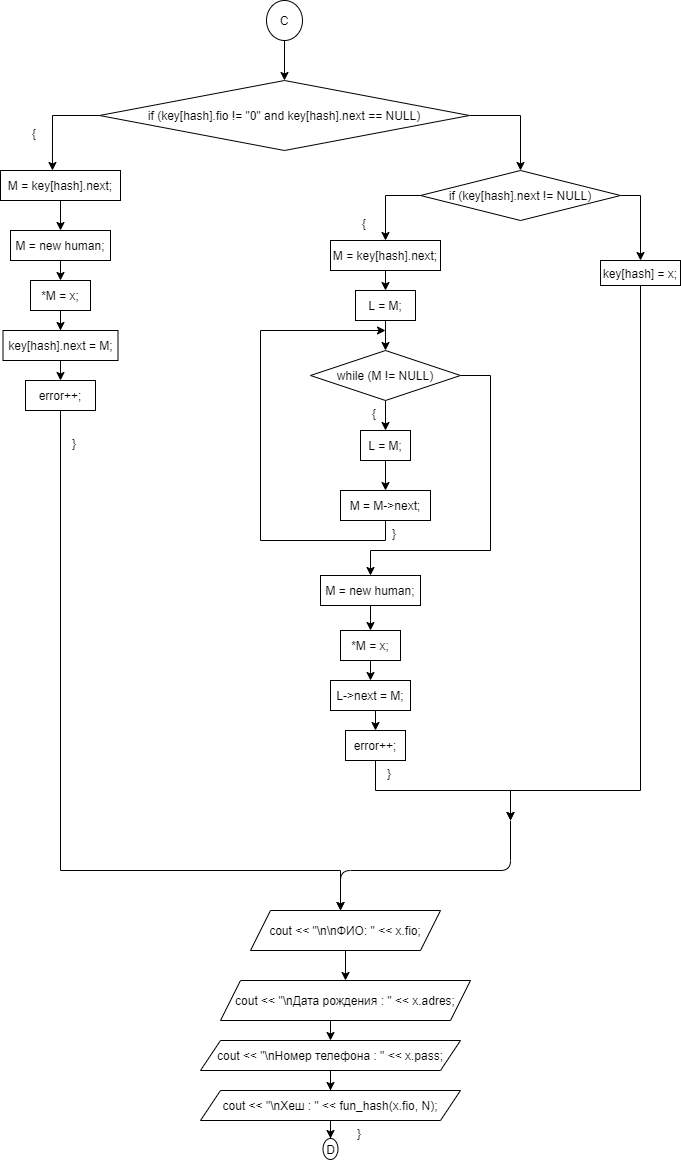
}

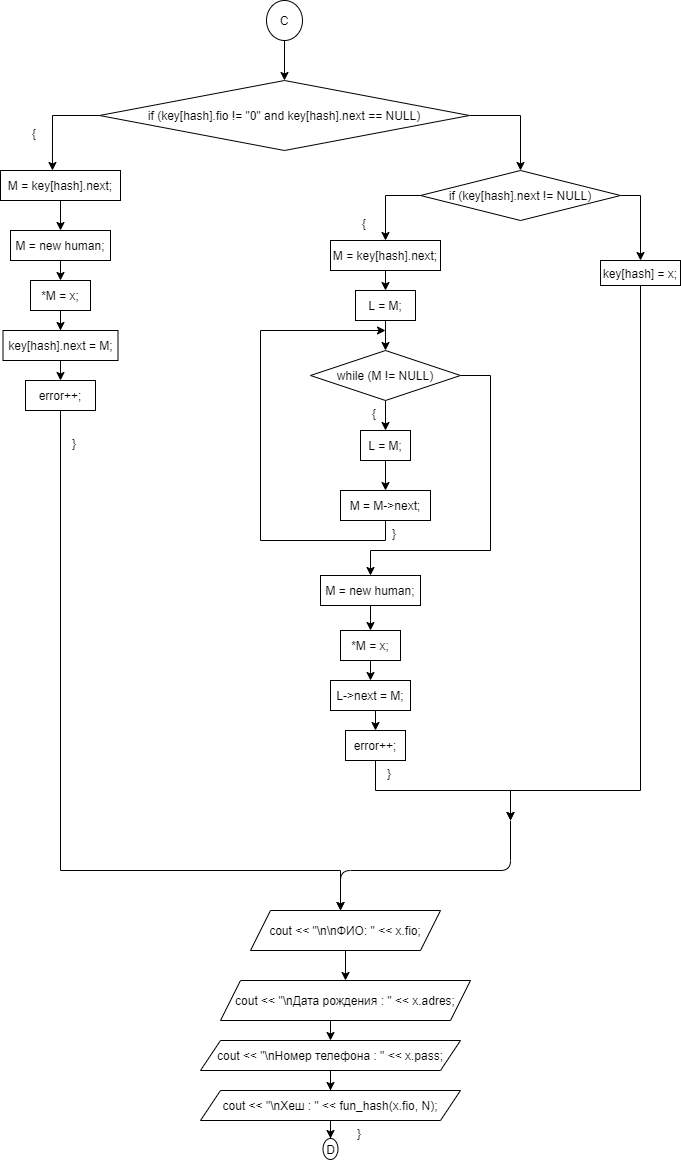
**Блок схема**

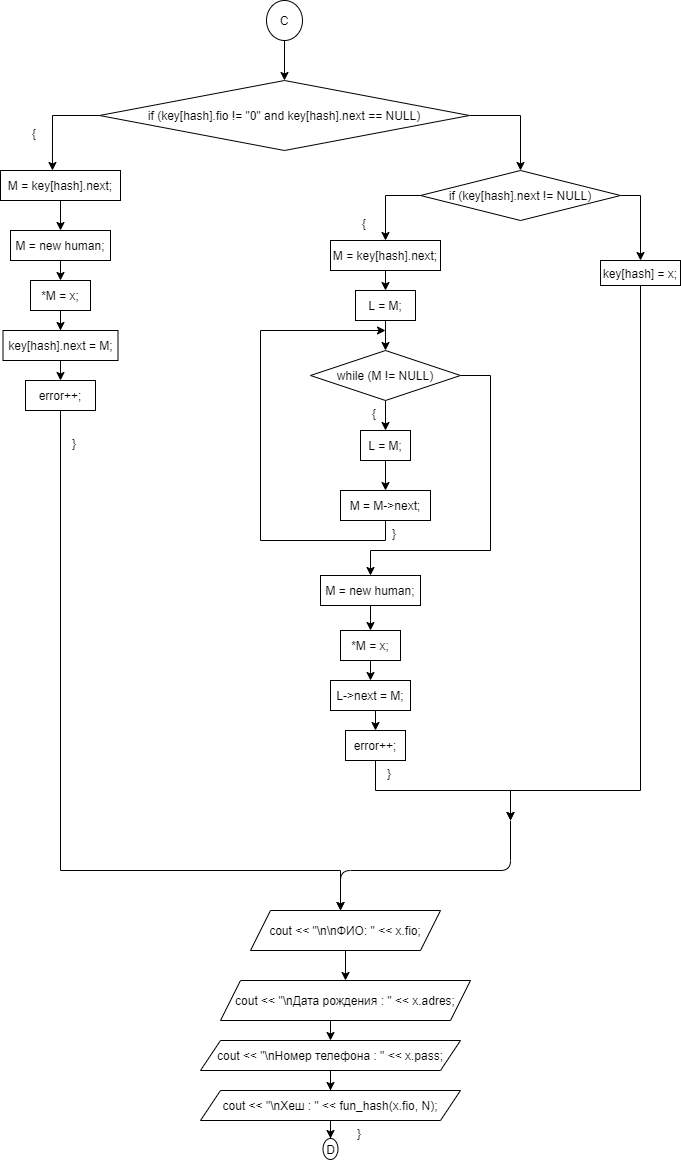


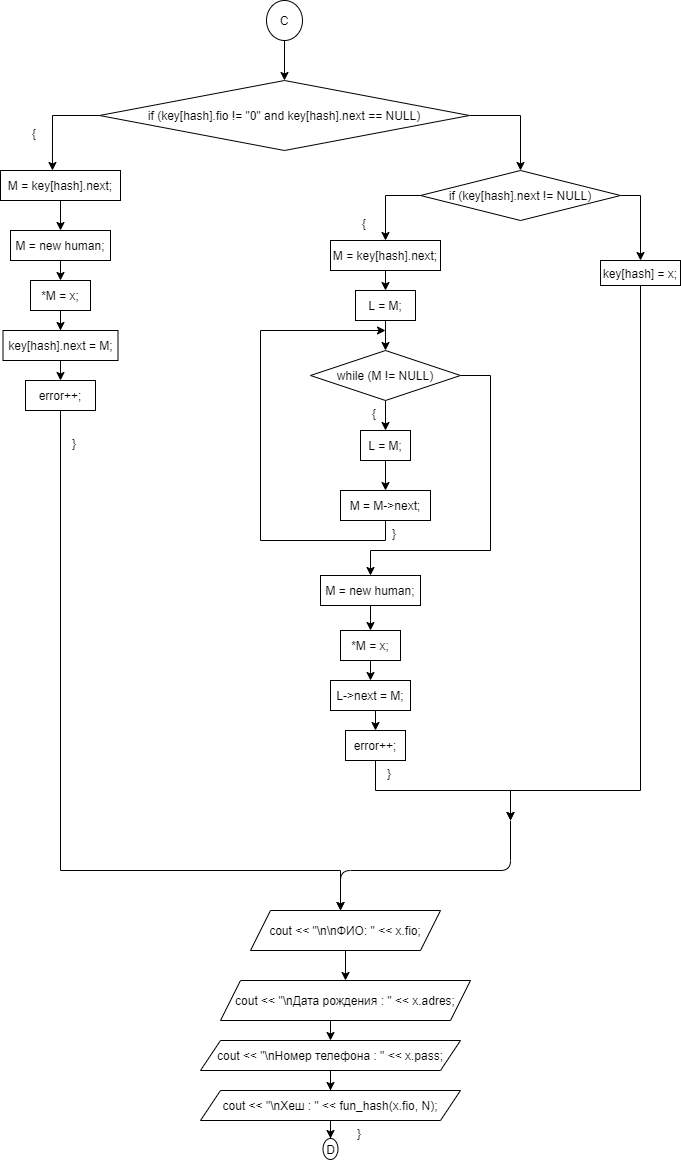












**Скриншоты тестов**

