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 №8**

Дисциплина: информатика

Тема: Структуры данных

Вариант: 25

Выполнил работу

студент группы РИС-20-1б

Рябов Никита Андреевич

Проверила

Доцент кафедры ИТАС Полякова О.А.

Пермь, 2021

**Цель**

Работа с двоичными файлами, организация ввода-вывода структурированной информации и ее хранение на внешних носителях.

**Постановка задачи**

Сформировать двоичный файл из элементов, заданной в варианте структуры, распечатать его содержимое, выполнить удаление и добавление элементов в соответствии со своим вариантом, используя для поиска удаляемых или добавляемых элементов функцию. Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

Структура "Студент":

* фамилия, имя, отчество;
* дата рождения;
* домашний адрес;
* рейтинг.

Удалить элементы, у которых даты рождения совпадают, добавить элемент перед элементом с заданной фамилией.

**Анализ задачи**

*Какие предстоит выполнить действия.*

Открыть файлы на чтение/запись

#include <fstream>

ifstream F1("F1.txt");

ofstream F2("F2.txt");

Считать данные. При применении fread/fwrite информация сохраняется в нечитаемой форме. Для удобства чтения и проверки работы программы было принято решение перейти на чтение через getline()

string x;

getline(F1,Data\_base[i].fio);

getline(F1,Data\_base[i].birthday);

getline(F1,Data\_base[i].adress);

getline(F1,x);

Data\_base[i].rating=stoi(x);

Получить от пользователя искомую фамилию

string find\_name;

cout << endl;

cout << "Введите искомую фамилию:\n";

cin >> find\_name;

cin.get();

Для сокращения расходов оперативной памяти операции удаления и добавления элементов объединены в один цикл. Удаление представляет собой запись только нужных элементов

Флажок, говорящий о наличии/отсутствии элементов с той же датой рождения.

bool flag=true;

for (int k=0;k<4;k++){

if (k!=i && Data\_base[i].birthday==Data\_base[k].birthday){

flag=false;

break;}}

Добавление нового элемента

if (NAME(Data\_base[i].fio)==find\_name)

{

string new\_fio;

string new\_birthday;

string new\_adress;

string new\_rating;

cout << "Обнаружена искомая фамилия\n";

cout << "Введите ФИО нового элемента\n";

getline(cin,new\_fio);

cout << "Введите дату рождения нового элемента\n";

getline(cin,new\_birthday);

cout << "Введите адрес нового элемента\n";

getline(cin,new\_adress);

cout << "Введите рейтинг нового элемента\n";

getline(cin,new\_rating);

F2<<new\_fio<<"\n";

F2<<new\_birthday<<"\n";

F2<<new\_adress<<"\n";

F2<<new\_rating<<"\n";

}

Функция NAME извлекает фамилию из строки, содержащей ФИО студента.

Запись текущего элемента, если необходимо.

if (flag)

{

F2<<Data\_base[i].fio<<"\n";

F2<<Data\_base[i].birthday<<"\n";

F2<<Data\_base[i].adress<<"\n";

F2<<Data\_base[i].rating<<"\n";

}

*С каким типом данных действуем*

Действуем с файлом. Файл хранит строки (string).

*В каком виде представлены данные*

Каждые 4 строки представляют собой информацию об одном студенте.

*Какие поля*

Структура Student имеет 4 поля. Поле fio типа string хранит ФИО студента. Поле birthday типа string хранит дату рождения студента. Поле adress типа string хранит адрес студента. Поле rating типа int хранит рейтинг студента.

*Операторы ввода/вывода*

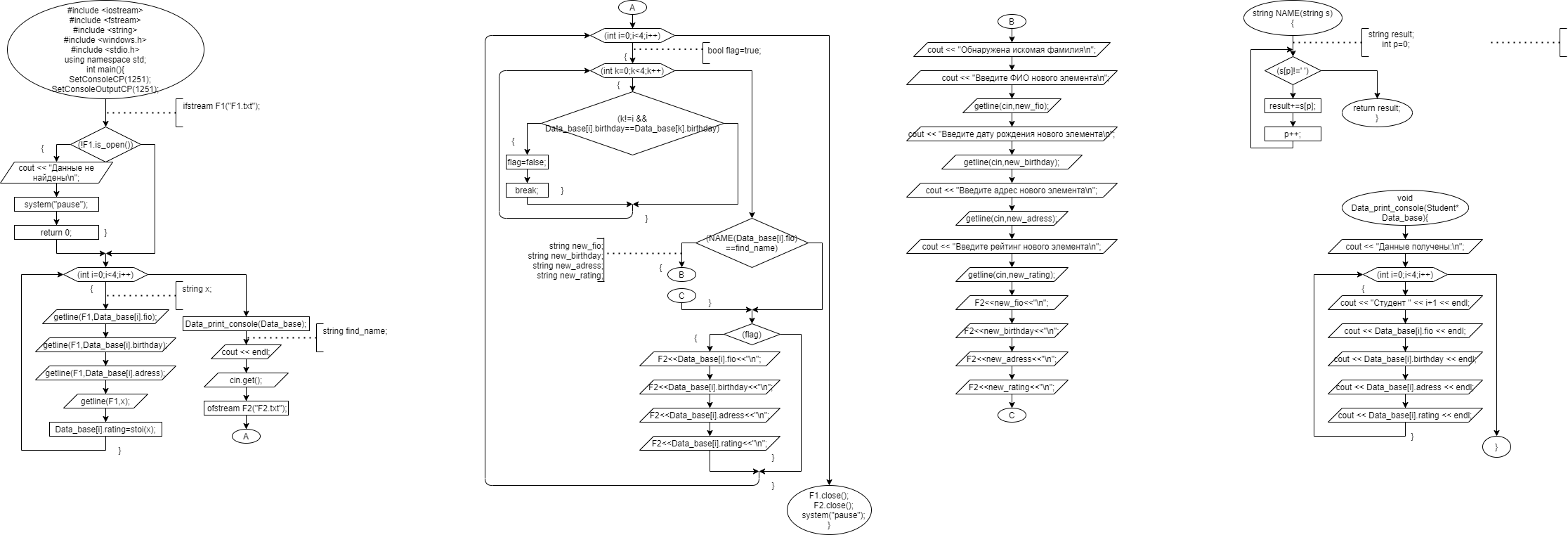
Ввод осуществляется командой getline(F1,row); где F1 – переменная файла, row – строковая переменная для записи данных.

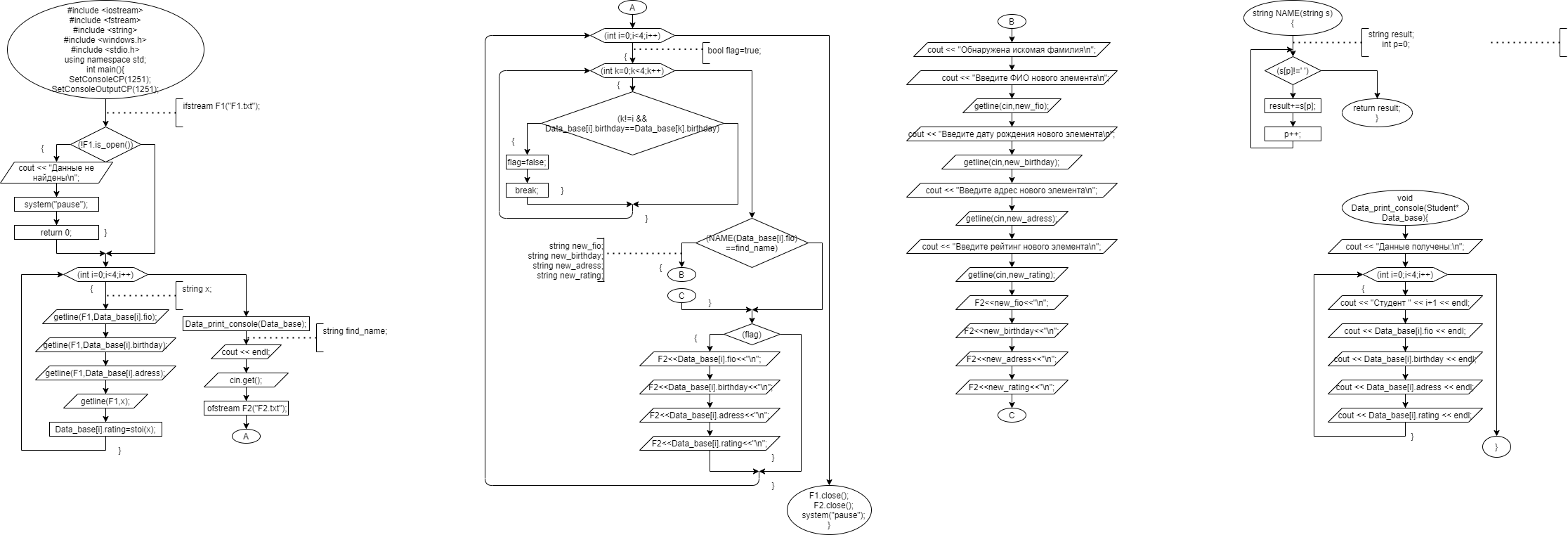
Вывод производится командой F2<<row<<"\n";

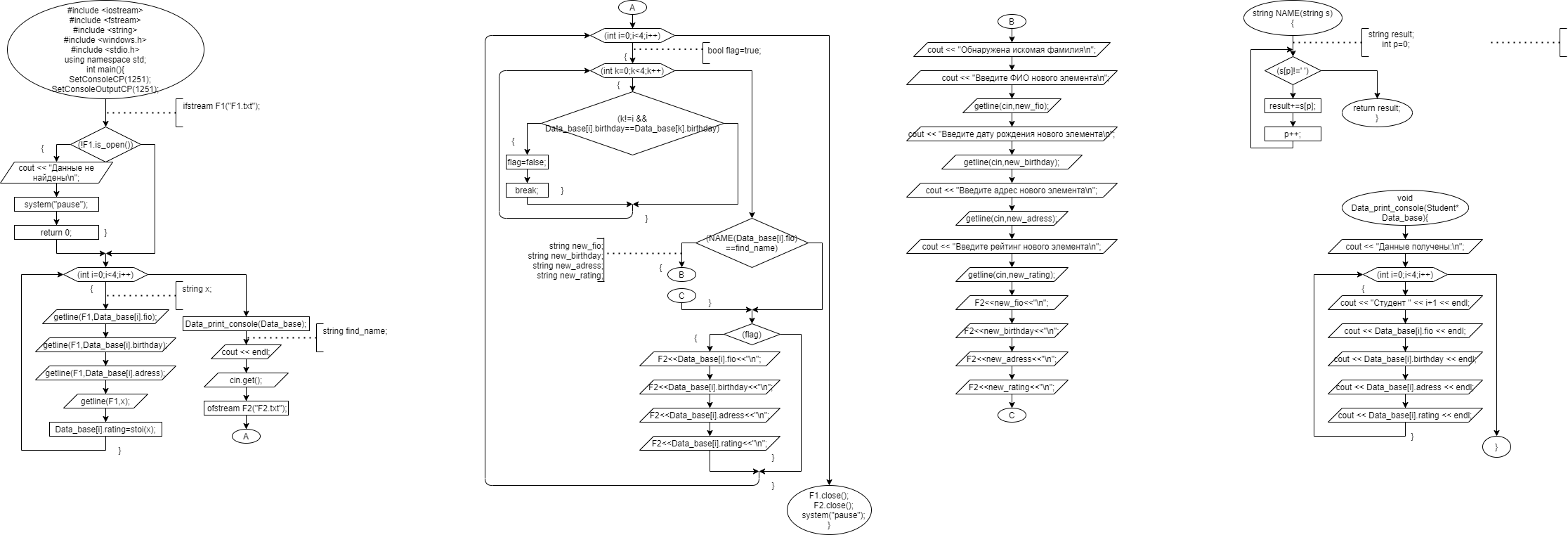
*Какие действия*

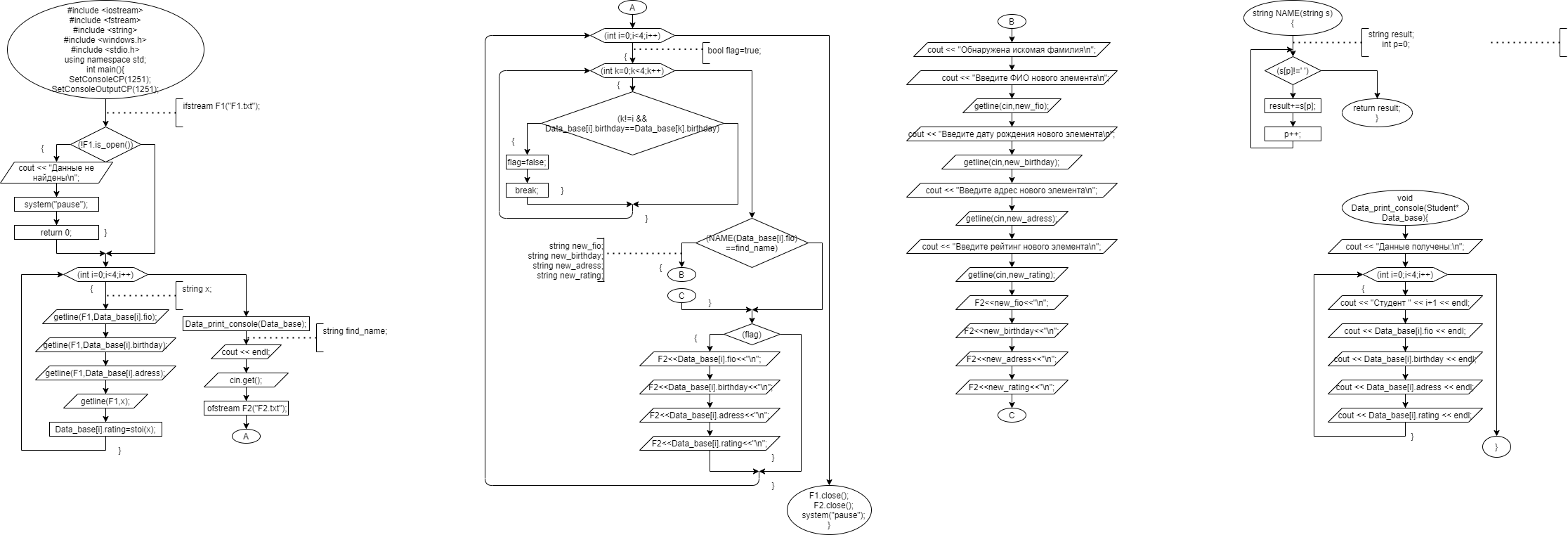
* Печать структур – for
* Извлечении фамилии из ФИО – while
* Проверка открытия файла – if
* Чтение данных – for
* Проверка корректности данных – try/catch
* Обработка структур – for
* Поиск дублей – for/if
* Проверка необходимости добавления элемента – if
* Проверка необходимости записи элемента в файл – if

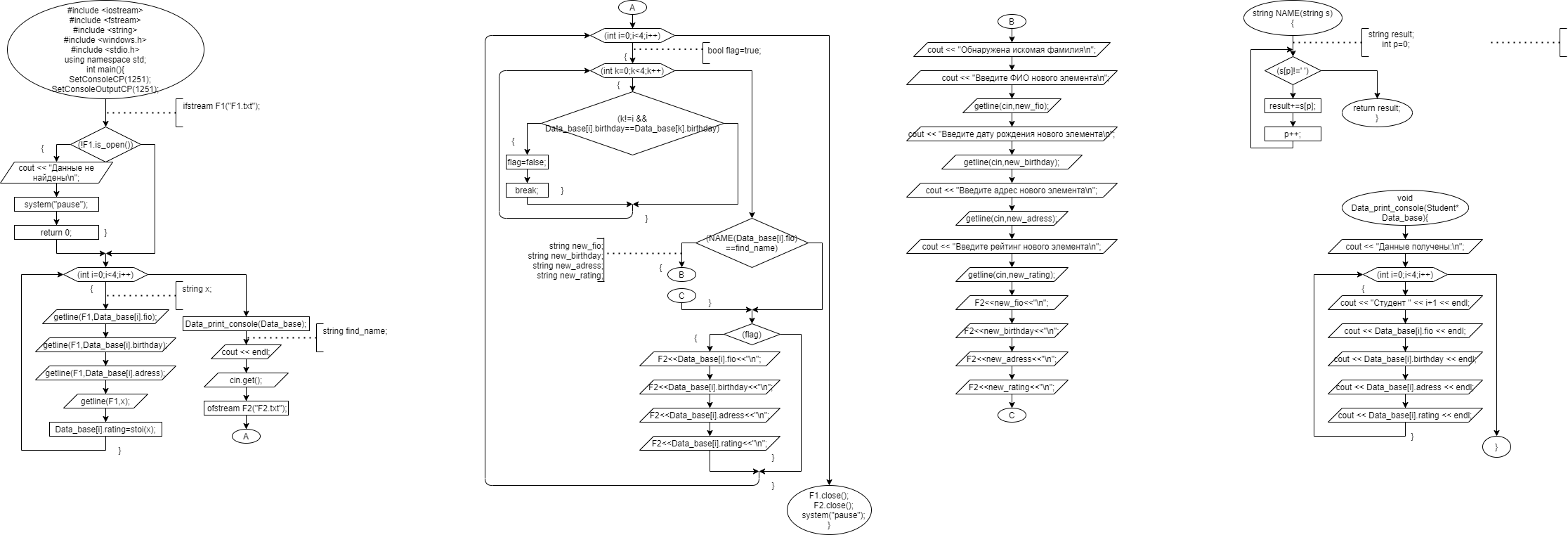
**Блок-схема**











**Код**

#include <iostream>

#include <fstream>

#include <string>

#include <windows.h>

#include <stdio.h>

using namespace std;

struct Student

{

string fio;

string birthday;

string adress;

int rating;

};

void Data\_print\_console(Student\* Data\_base)

{

cout << "Данные получены:\n";

for (int i=0;i<4;i++)

{

cout << "Студент " << i+1 << endl;

cout << Data\_base[i].fio << endl;

cout << Data\_base[i].birthday << endl;

cout << Data\_base[i].adress << endl;

cout << Data\_base[i].rating << endl;

}

}

string NAME(string s)

{

string result;

int p=0;

while (s[p]!=' ')

{

result+=s[p];

p++;

}

return result;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Student Data\_base[4];

ifstream F1("F1.txt");

if (!F1.is\_open())

{

cout << "Данные не найдены\n";

system("pause");

return 0;

}

try

{

for (int i=0;i<4;i++)

{

string x;

getline(F1,Data\_base[i].fio);

getline(F1,Data\_base[i].birthday);

getline(F1,Data\_base[i].adress);

getline(F1,x);

Data\_base[i].rating=stoi(x);

}

}

catch(...)

{

cout << "Данные повреждены или не соответствуют формату\n";

system("pause");

return 0;

}

Data\_print\_console(Data\_base);

string find\_name;

cout << endl;

cout << "Введите искомую фамилию:\n";

cin >> find\_name;

cin.get();

ofstream F2("F2.txt");

for (int i=0;i<4;i++)

{

bool flag=true;

for (int k=0;k<4;k++)

{

if (k!=i && Data\_base[i].birthday==Data\_base[k].birthday)

{

flag=false;

break;

}

}

if (NAME(Data\_base[i].fio)==find\_name)

{

string new\_fio;

string new\_birthday;

string new\_adress;

string new\_rating;

cout << "Обнаружена искомая фамилия\n";

cout << "Введите ФИО нового элемента\n";

getline(cin,new\_fio);

cout << "Введите дату рождения нового элемента\n";

getline(cin,new\_birthday);

cout << "Введите адрес нового элемента\n";

getline(cin,new\_adress);

cout << "Введите рейтинг нового элемента\n";

getline(cin,new\_rating);

F2<<new\_fio<<"\n";

F2<<new\_birthday<<"\n";

F2<<new\_adress<<"\n";

F2<<new\_rating<<"\n";

}

if (flag)

{

F2<<Data\_base[i].fio<<"\n";

F2<<Data\_base[i].birthday<<"\n";

F2<<Data\_base[i].adress<<"\n";

F2<<Data\_base[i].rating<<"\n";

}

}

F1.close();

F2.close();

system("pause");

}

**Скриншот**

