Министерство образования и науки Российской Федерации

Федеральное государственное образовательное автономное учреждение высшего образования

"Пермский национальный исследовательский политехнический университет"

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

Дисциплина: Основы алгоритмизации и программирования

Тема: Блоковый ввод вывод

|  |  |
| --- | --- |
| Выполнил работу | |
| Студент группы РИС-22-1б | |
| Деревнин И.В. | |
|  | |
| Проверил работу | |
| Доцент кафедры ИТАС | |
| Полякова О.А. | |
|  | |

Пермь – 2023

**Анализ предметной области**

**Постановка задачи**

Реализовать структуру "Студент":

* фамилия, имя, отчество;
* номер телефона;
* группа;
* оценки по 3 основным предметам.

Реализовать функции, которые взаимодействуют с данной структурой. Удаление всех элементов из группы с указанным номером, у которых среднее арифметическое оценок меньше заданного, добавление элемента после элемента с заданной фамилией, заполнение структуры.

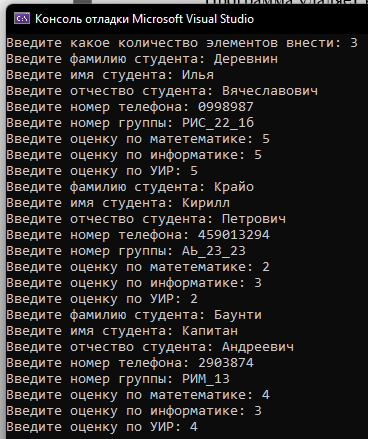
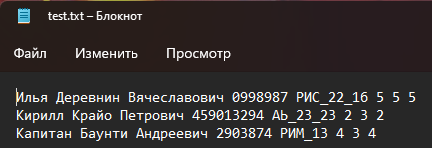
**Анализ задачи**

Для того чтобы выполнить поставленную задачу, необходимо реализовать структуру со всеми полями, которые содержатся в задаче. Для данной структуры реализовать функции, указанные в задании. Для этого будут использоваться библиотеки: fstream -для работы с файлами, записи и считывания данных, windows.h – для корректного считывания данных вводимых из консоли. Для открытия и создания файлов будет использован метод open класса fstream, вызываемые через объект данного класса, также будет использоваться метод is\_open, который проверяет на корректность, создание или открытие файла. Функция SetConsoleCP(1251) с параметром 1251 – используется для того, чтобы получать корректные данные русской локализации из консоли и вписывать в файл. Метод close класса fstream нужен для корректного закрытия и сохранения файла. Далее будут созданы функции: input - функция которая заполняет файл введенными из консоли данными, clear\_stud – функция которая удаляет из файла данные тех студентов, у которых средний бал по предметам ниже указанного нами в ручную, add – функция, которая выполняет поиск по фамилии и после добавляет еще один элемент структуры «Студент» в файл.

Для того чтобы протестировать программу необходимо: протестировать работу функции input, clear\_student и add, а после сравнить с результатами, записанными в файл.

**Тестирование программ**

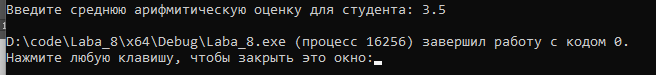
Функция input:

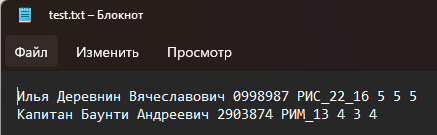
*Рис. 1 – Занесение данных в файл через функцию input.*

*Рис. 2 – Результат выполнения программы в файле.*

Функция clear\_student:

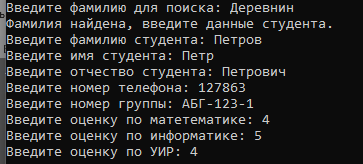
Исходный файл для очистки представлен на рисунке 2.

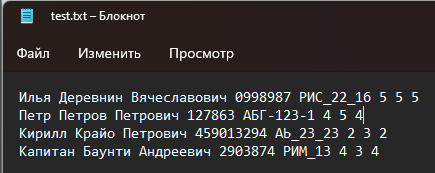
*Рис. 3 – Результат выполнения программы в консоли.*

*Рис. 4 – Результат работы функции в файле.*

Функция add:

Исходный файл для добавления представлен на рисунке 2.

*Рис. 5 – Поиск по фамилии и занесение данных в файл.*

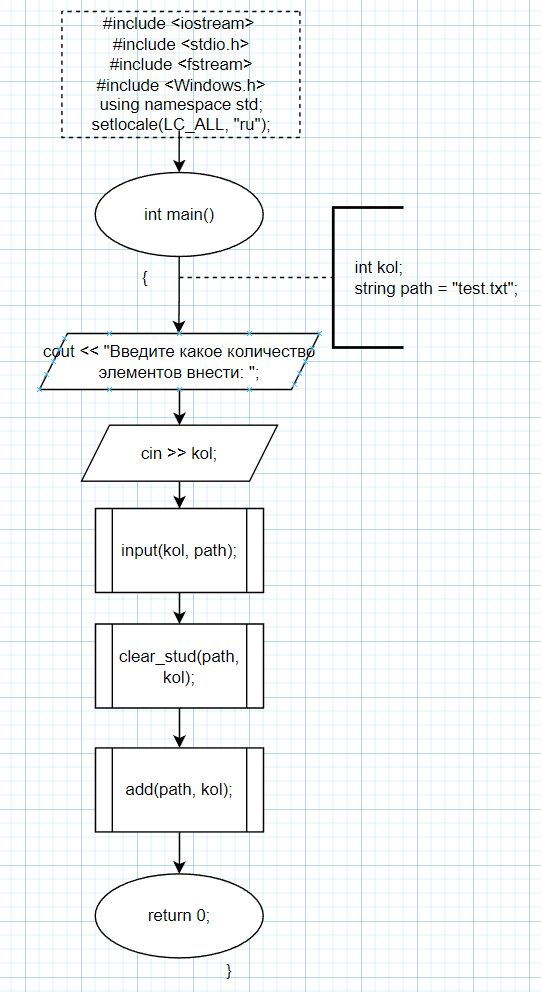
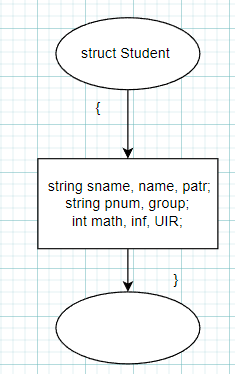
*Рис. 6 – Результат работы функции в файле.*

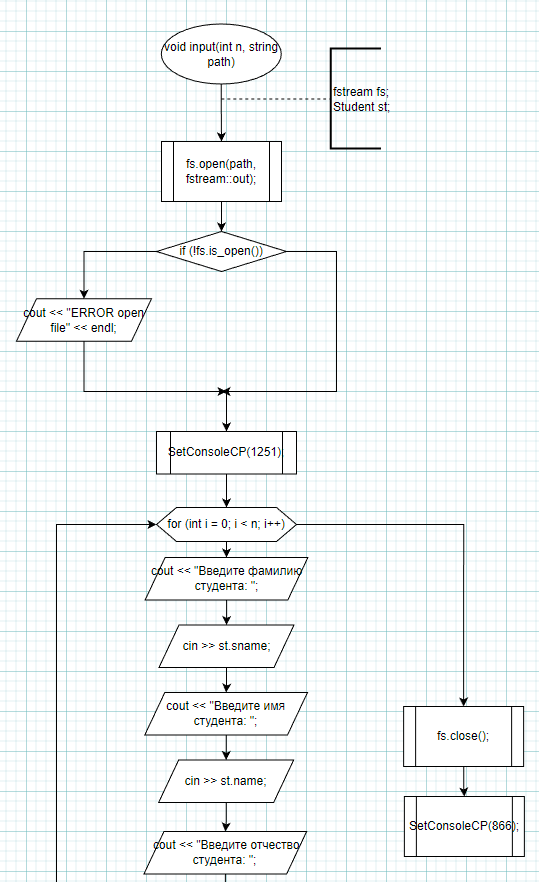
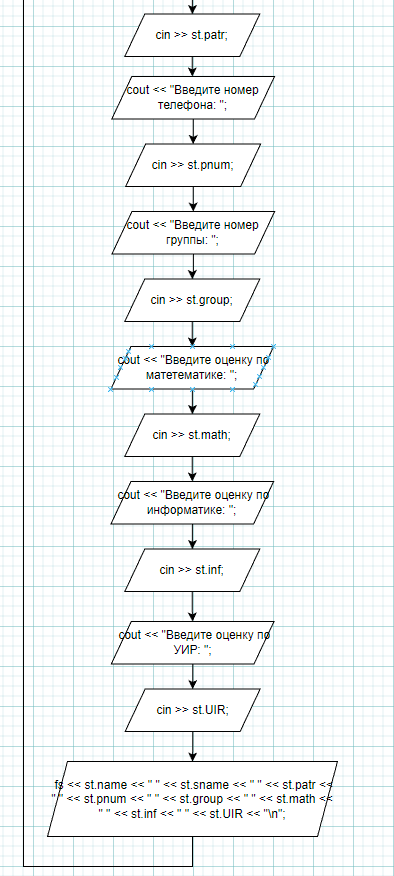
**Заключение**

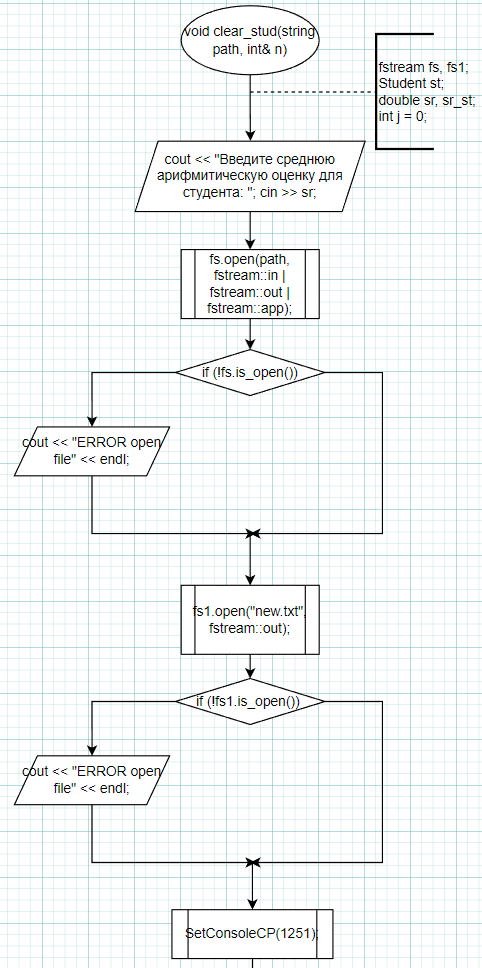
Была разработана программа, которая работает с файлами и структурой. В программе реализованы функции для заполнения файла по структуре, считывание и удаление из файла студентов по определенным критериям, а также поиск по файлу, для дальнейшего добавления строчек в файл. Данная программа работает стабильно и корректно, но для большего удобства нужно реализовать меню программы.

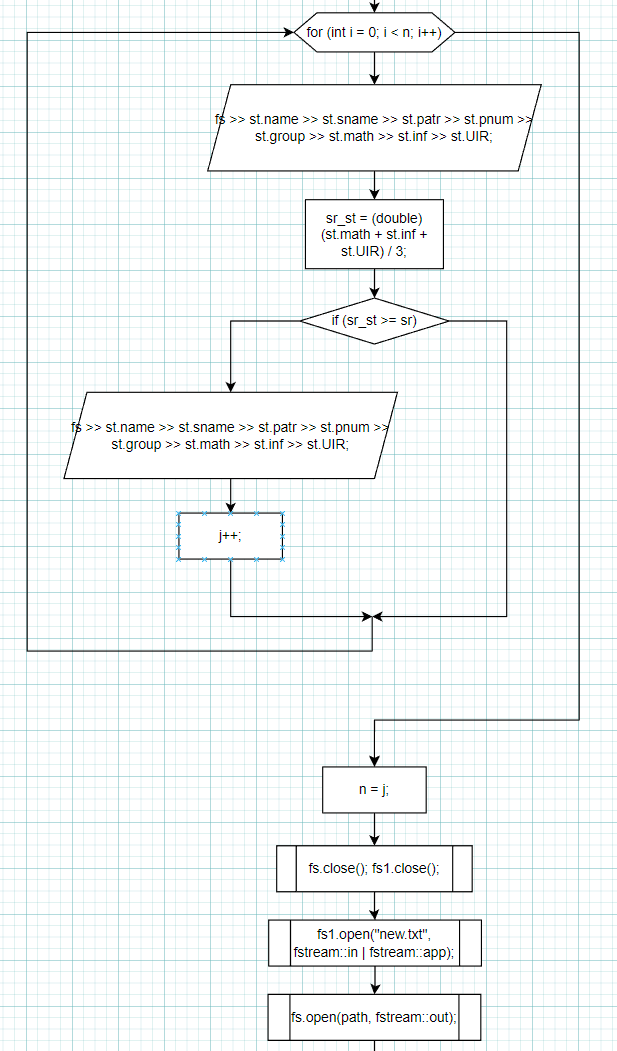
**Приложения**

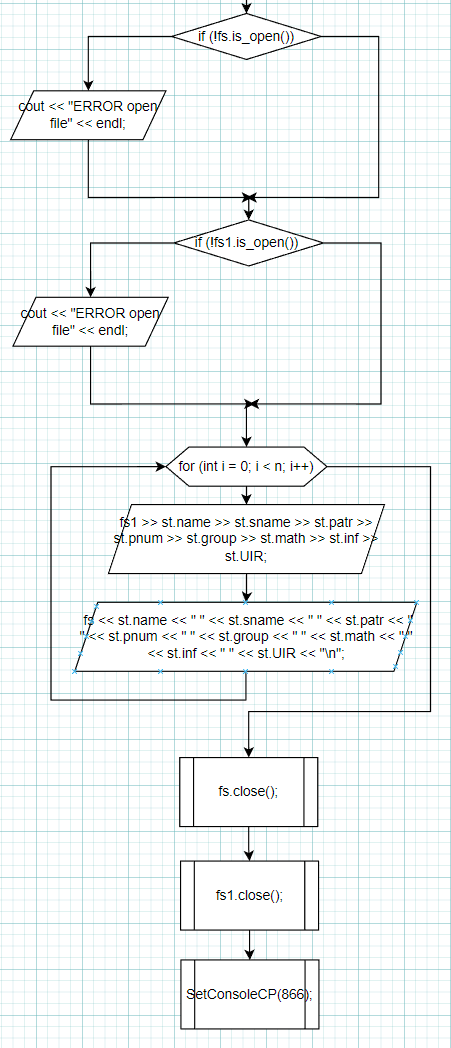
Приложение А

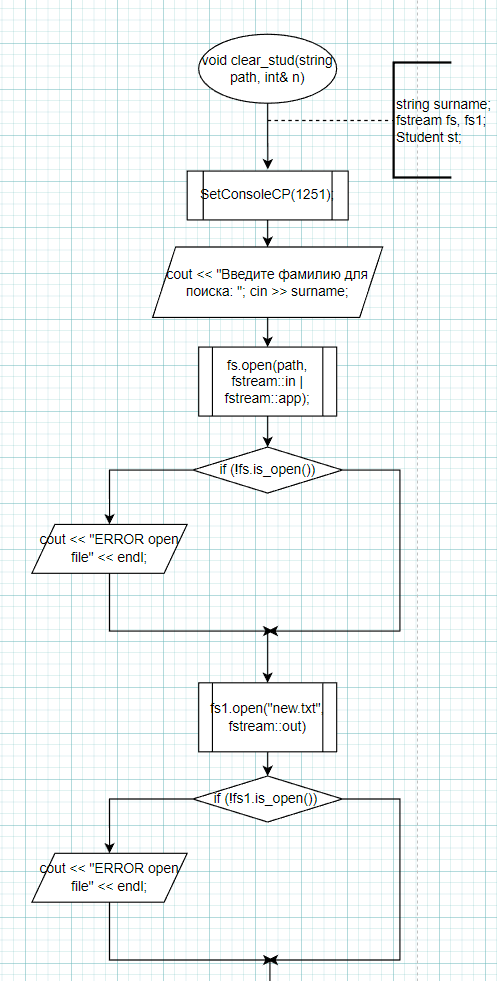
 

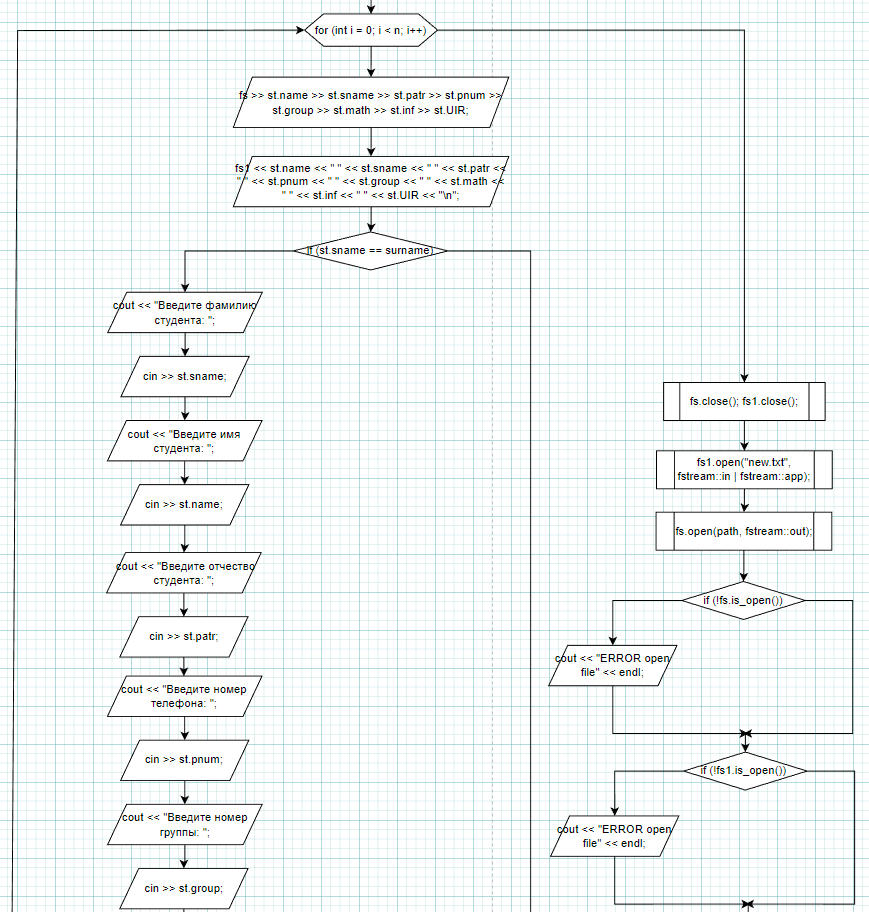
 

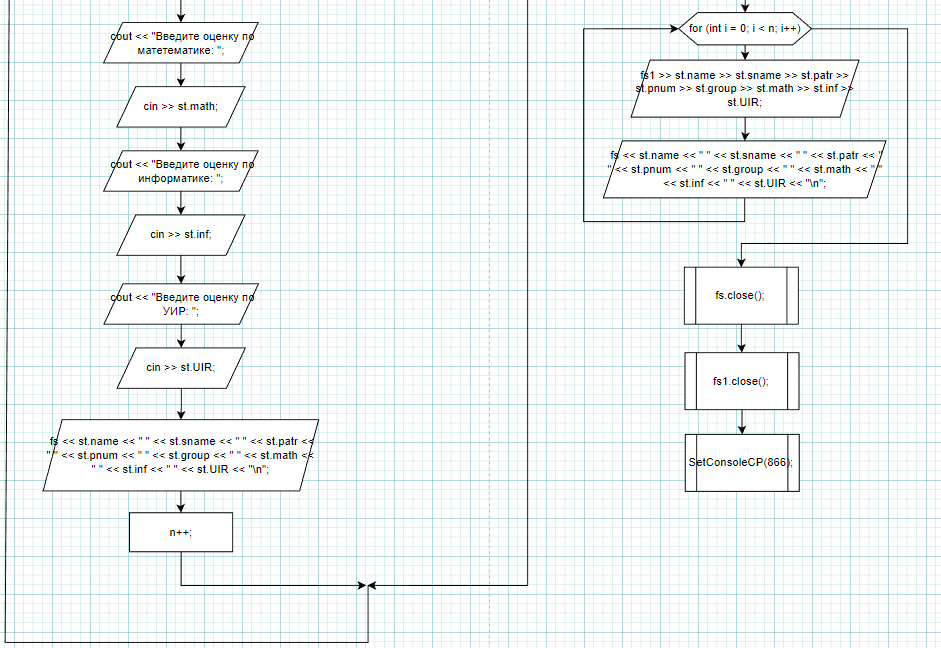












Приложение Б.

#include <iostream> //24 вариант

#include <stdio.h>

#include <fstream>

#include <Windows.h>

using namespace std;

struct Student //объявление структуры

{

string sname, name, patr; //поля структуры

string pnum, group;

int math, inf, UIR;

};

void input(int n, string path) //функция добавления в файл

{

fstream fs; //создание потока ввода/вывода

Student st; //создание структуры для шаблонного заполнения

fs.open(path, fstream::out); //открытие/создание файла для записи туда

значений

if (!fs.is\_open()) //проверка на открытие

{

cout << "ERROR open file" << endl;

}

SetConsoleCP(1251); //изменение кодировки консоли для корректной

записи в файл

for (int i = 0; i < n; i++) //получение данных и запись в файл

{

cout << "Введите фамилию cтудента: "; cin >> st.sname;

cout << "Введите имя cтудента: "; cin >> st.name;

cout << "Введите отчество cтудента: "; cin >> st.patr;

cout << "Введите номер телефона: "; cin >> st.pnum;

cout << "Введите номер группы: "; cin >> st.group;

cout << "Введите оценку по матетематике: "; cin >> st.math;

cout << "Введите оценку по информатике: "; cin >> st.inf;

cout << "Введите оценку по УИР: "; cin >> st.UIR;

fs << st.name << " " << st.sname << " " << st.patr << " " << st.pnum << " " << st.group << " " << st.math << " " << st.inf << " " << st.UIR << "\n";

}

fs.close(); //закрытие файла

SetConsoleCP(866); //изменение кодировки консоли

}

void clear\_stud(string path, int& n) //функция для отсеивания студентов по среднему баллу

{

fstream fs, fs1;

Student st;

double sr, sr\_st;

cout << "Введите среднюю арифмитическую оценку для студента: "; cin >> sr;

fs.open(path, fstream::in | fstream::out | fstream::app); //открытие файла без удаления, для считывания и записи

if (!fs.is\_open()) //блок проверки

{

cout << "ERROR open first file" << endl;

}

fs1.open("new.txt", fstream::out); //открытие файла для записи

if (!fs1.is\_open()) //блок проверки на правильность открытия файла

{

cout << "ERROR open second file" << endl;

}

SetConsoleCP(1251);

int j = 0;

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

{

fs >> st.name >> st.sname >> st.patr >> st.pnum >> st.group >> st.math >> st.inf >> st.UIR; //получение данных из файла

sr\_st = (double)(st.math + st.inf + st.UIR) / 3; //вычисление ср. оценки

if (sr\_st >= sr)

{

fs1 << st.name << " " << st.sname << " " << st.patr << " " << st.pnum << " " << st.group << " " << st.math << " " << st.inf << " " << st.UIR << "\n";

j++; //запись в другой файл

}

}

n = j;

fs.close(); fs1.close(); //закрытие файлов для дальнейшего корректного открытия

fs1.open("new.txt", fstream::in | fstream::app); //открытие файла без удаления для считывания

fs.open(path, fstream::out); //открытие пустого файла для записи

if (!fs.is\_open()) //блок проверки на корректность открытия

{

cout << "ERROR open first file" << endl;

}

if (!fs1.is\_open())

{

cout << "ERROR open second file" << endl;

}

for (int i = 0; i < n; i++) //блок переписывания файлов

{

fs1 >> st.name >> st.sname >> st.patr >> st.pnum >> st.group >> st.math >> st.inf >> st.UIR;

fs << st.name << " " << st.sname << " " << st.patr << " " << st.pnum << " " << st.group << " " << st.math << " " << st.inf << " " << st.UIR << "\n";

}

fs.close(); //закрытие файлов

fs1.close();

SetConsoleCP(866);

}

void add(string path, int &n) //функция для добавления после заданной фамилии.

{

string surname;

SetConsoleCP(1251);

cout << "Введите фамилию для поиска: "; cin >> surname;

fstream fs, fs1; //создание потоков для работы с файлами

Student st; //создание структуры как шаблона

fs.open(path, fstream::in | fstream::app); //открытие файла для считывания без удаления

if (!fs.is\_open())

{

cout << "ERROR open first file" << endl;

}

fs1.open("new.txt", fstream::out); //открытие файла для записи

if (!fs1.is\_open())

{

cout << "ERROR open second file" << endl;

}

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

{

fs >> st.name >> st.sname >> st.patr >> st.pnum >> st.group >> st.math >> st.inf >> st.UIR; //считывание из исходного файла

fs1 << st.name << " " << st.sname << " " << st.patr << " " << st.pnum << " " << st.group << " " << st.math << " " << st.inf << " " << st.UIR << "\n";

if (st.sname == surname) //записывание в файл (сверху)

{ //это блок добавления нового студента в файл

cout << "Фамилия найдена, введите данные студента." << endl;

cout << "Введите фамилию cтудента: "; cin >> st.sname;

cout << "Введите имя cтудента: "; cin >> st.name;

cout << "Введите отчество cтудента: "; cin >> st.patr;

cout << "Введите номер телефона: "; cin >> st.pnum;

cout << "Введите номер группы: "; cin >> st.group;

cout << "Введите оценку по матетематике: "; cin >> st.math;

cout << "Введите оценку по информатике: "; cin >> st.inf;

cout << "Введите оценку по УИР: "; cin >> st.UIR;

fs1 << st.name << " " << st.sname << " " << st.patr << " " << st.pnum << " " << st.group << " " << st.math << " " << st.inf << " " << st.UIR << "\n";

n++;

}

}

fs.close(); fs1.close(); //закрытие файлов для дальнейшего коректного открытия

fs1.open("new.txt", fstream::in | fstream::app); //открытие файла для чтения без удаления

fs.open(path, fstream::out); //открытие чистого файла для записи

if (!fs.is\_open())

{

cout << "ERROR open first file" << endl;

}

if (!fs1.is\_open())

{

cout << "ERROR open second file" << endl;

}

for (int i = 0; i < n; i++) //переписывание данных из одного файла в другой

{

fs1 >> st.name >> st.sname >> st.patr >> st.pnum >> st.group >> st.math >> st.inf >> st.UIR;

fs << st.name << " " << st.sname << " " << st.patr << " " << st.pnum << " " << st.group << " " << st.math << " " << st.inf << " " << st.UIR << "\n";

}

fs.close(); //закрытие файлов

fs1.close();

SetConsoleCP(866);

}

int main()

{

setlocale(LC\_ALL, "ru");

int kol;

string path = "test.txt"; //задание названия исходного файла

cout << "Введите какое количество элементов внести: ";

cin >> kol;

input(kol, path); //вызов функции заполнения

clear\_stud(path, kol); //вызов функции сортировки

add(path, kol); //вызов функции добавления

return 0;

}