Министерство высшего образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

по лабораторной работе №8

Тема: «Блоковый ввод - вывод»

Выполнил

Студент группы РИС-22-2б

Будин Д.В.

Проверил доц. Кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2023

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.

24 ВАРИАНТ.

Структура "Студент":

- фамилия, имя, отчество;

- номер телефона;

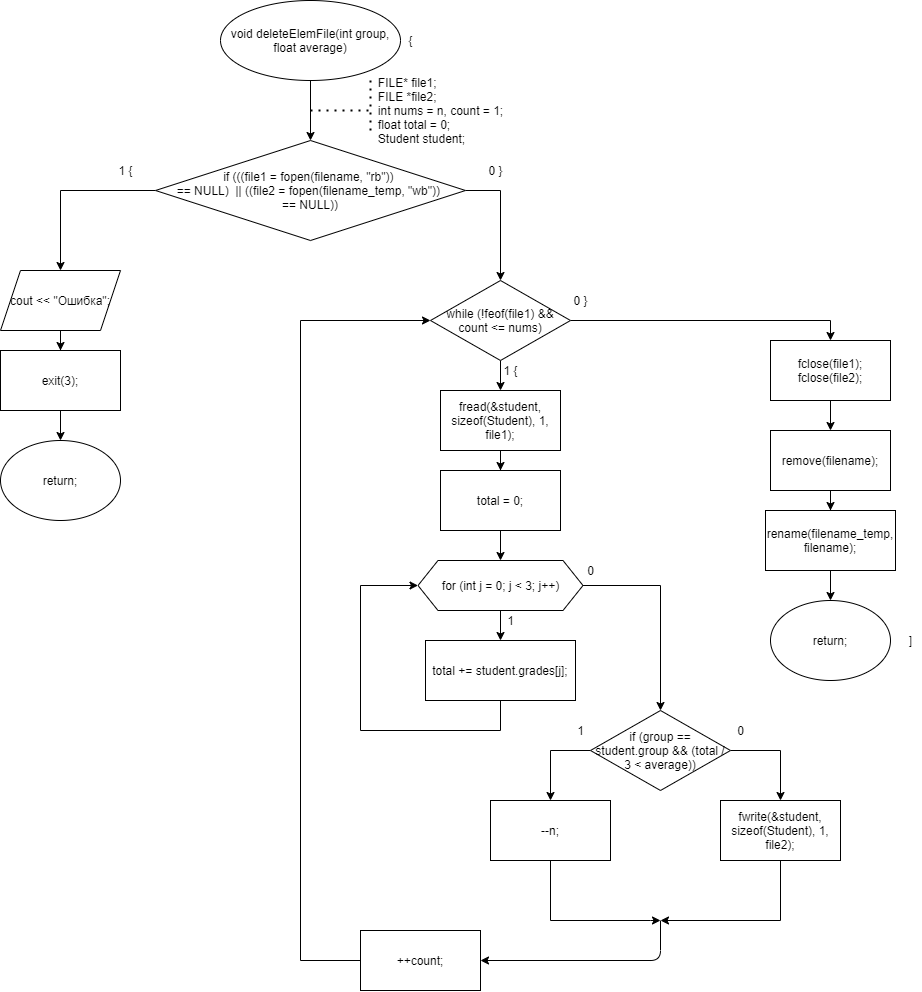
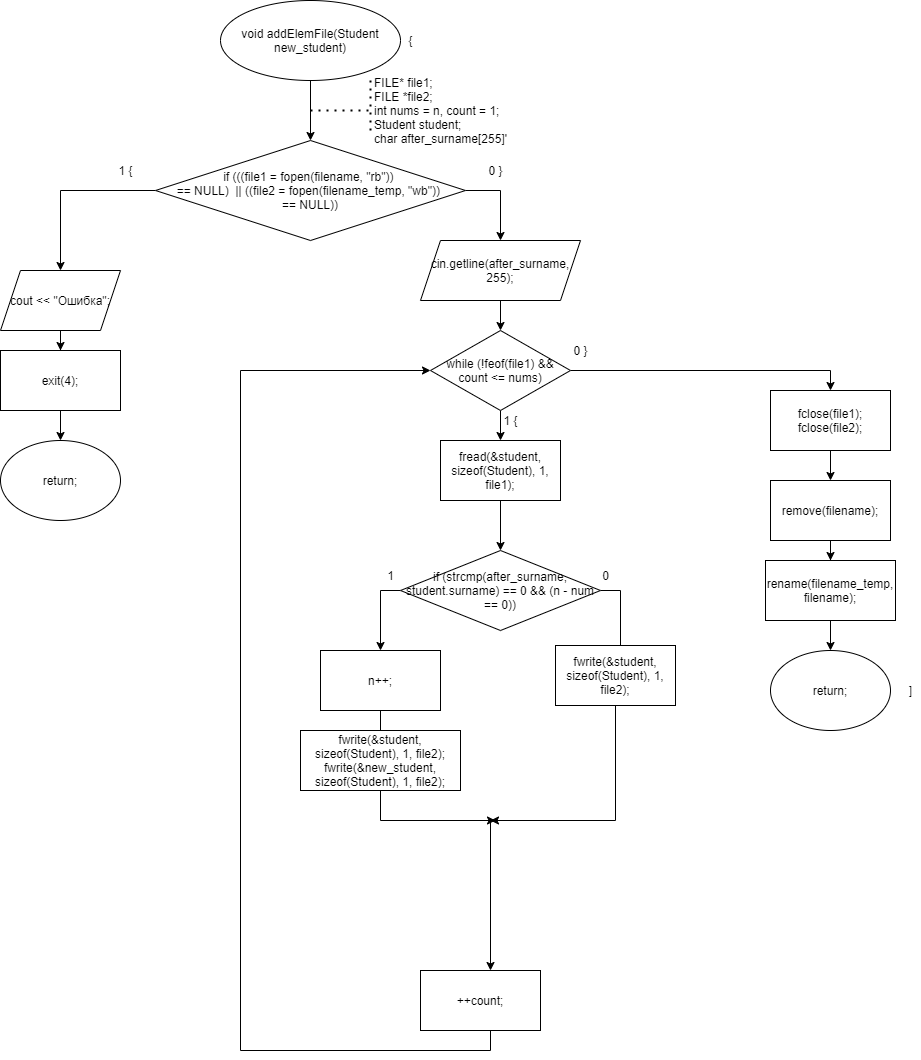
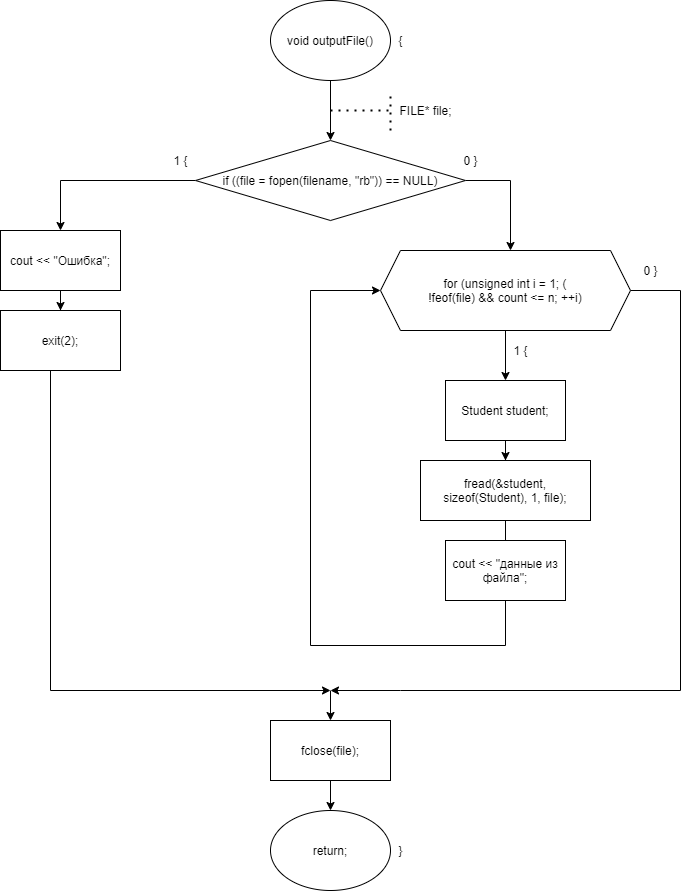
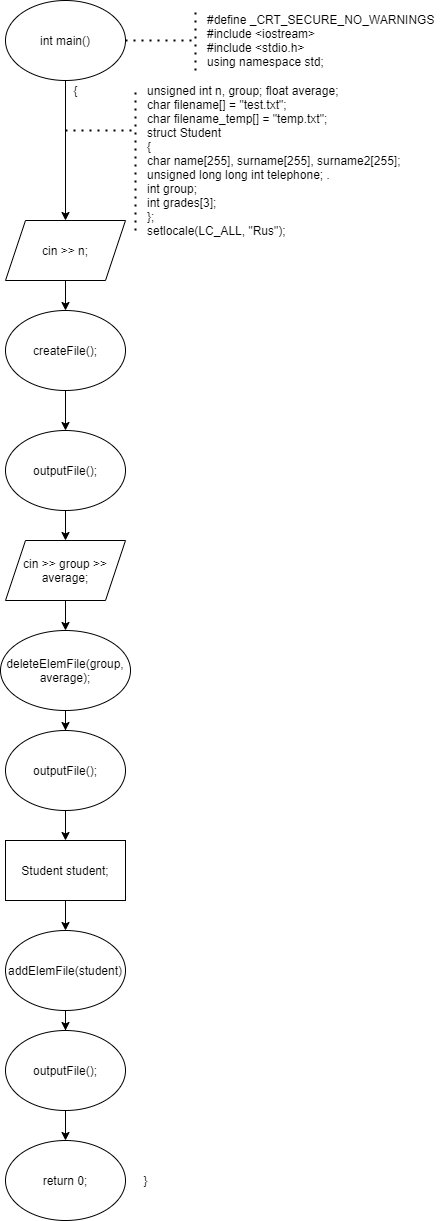
- группа;

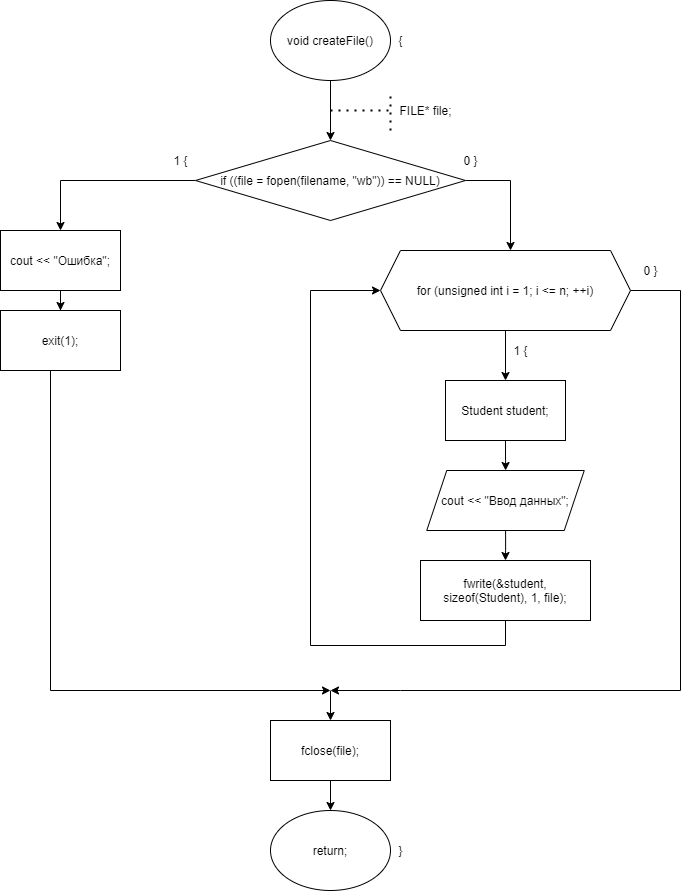
- оценки по 3 основным предметам.

Удалить все элементы из группы с указанным номером, у которых среднее арифметическое оценок меньше заданного, добавить элемент после элемента с заданной фамилией.

Анализ задачи

1. Формируем глобальные переменные для общей видимости: число студентов unsigned int n, структуру Student, строковые переменные filename и filename\_temp, хранящие названия файлов для основного и буферного файла соответственно.
2. Вводим с клавиатуры число студентов для создания n-структур и их записи в двоичный файл через ф-ию createFile: if ((file = fopen(filename, "wb")) == NULL) – создание/открытие файла, если проходит проверку. С клавиатуры вводятся данные пользователя и записываются в файл: fwrite(&student, sizeof(Student), 1, file);. Файл закрывается.
3. Вывод данных n-структур с помощью функции outputFile (чтение файла): if ((file = fopen(filename, "rb")) == NULL) – создание/открытие файла, если проходит проверку. Файл считывается, происходит вывод содержимого. Файл закрывается.
4. Ввод двух значений в переменные group и average (по заданию): группа и среднее арифметическое оценок. Данные передаются в ф-ию deleteElemFile(int group, float average);
5. if (((file1 = fopen(filename, "rb")) == NULL) || ((file2 = fopen(filename\_temp, "wb")) == NULL)) – проверка открытия/создания файла. Проход по структурам основного файла и if (group == student.group && (total / 3 < average)), то –n – уменьшение кол-во студентов на 1, иначе пользователь записывается в буферный файл.
6. После оба файла закрываются, основной файл удаляем, буферный файл переименуем в основной. Вывод файла через outputFile.
7. Для добавления нового пользователя после заданной фамилии формируем новый объект student структуры Student и передаем ее в ф-ию addElemFile(Student new\_student)
8. if (((file1 = fopen(filename, "rb")) == NULL) || ((file2 = fopen(filename\_temp, "wb")) == NULL)) – проверка открытия/создания файла. Проходим по основному файла и if (strcmp(after\_surname, student.surname) == 0 && (n - num == 0)), где strcmp – сравниваем строки, а n - num == 0 проверяется на первое вхождение фамилии, то записывается данные из основного файла, а после записывается заданный пользователь, n++. Иначе в буферный файл записывается пользователь из основного файла.
9. Закрываем файлы, удаляем основной файл, переименуем буферный файл в основной. Вывод файла через outputFile.

**Блок-схемы **

****

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

using namespace std;

unsigned int n; //кол-во учеников, считывание в main

char filename[] = "test.txt";

char filename\_temp[] = "temp.txt";

struct Student

{

char name[255], surname[255], surname2[255];

unsigned long long int telephone; //11-значное число, нужно больше места.

int group;

int grades[3];

};

void createFile() //Создание БД с n-структурами Student;

{

FILE\* file; //указатель на файл

if ((file = fopen(filename, "wb")) == NULL) //ошибка открытия файла

{

cout << "ЧЕТ НЕ ТО" << endl;

return;

}

for (unsigned int i = 1; i <= n; ++i) //цикл по студентам

{

Student student;

cout << "Введите ФИО: " << endl;

cin.get();

cout << "\* Введите фамилию: ";

cin.getline(student.surname, 255);

cout << "\* Введите имя: ";

cin.getline(student.name, 255);

cout << "\* Введите отчество: ";

cin.getline(student.surname2, 255);

cout << "Введите номер телефона: ";

cin >> student.telephone;

cout << "Введите номер группы: ";

cin.get();

cin >> student.group;

cout << "Введите 3 основных оценки через пробел: ";

for (int i = 1; i <= 3; i++) cin >> student.grades[i - 1];

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

fwrite(&student, sizeof(Student), 1, file); //запись структуры

}

fclose(file); //закрываем файл

return;

}

void outputFile() //вывод содержимого файла

{

FILE\* file;

if ((file = fopen(filename, "rb")) == NULL)

{

cout << "ЧЕТ НЕ ТО" << endl;

return;

}

unsigned int count = 1;

while (!feof(file) && count <= n)

{

Student student;

fread(&student, sizeof(Student), 1, file);

cout << "ФИО: " << student.surname << " " << student.name << " " << student.surname2 << endl;

cout << "Телефон: " << student.telephone << endl;

cout << "Группа: " << student.group << endl;

cout << "Оценки(3 основных): ";

for (int j = 0; j < 3; j++)

cout << student.grades[j] << " ";

cout << endl << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

if (ferror(file)) {

perror("Read error");

break;

}

++count;

}

fclose(file);

return;

} //вывод содержимого файла //

void deleteElemFile(int group, float average) //удаление элементов с average(grades) < average заданного

{

FILE\* file1;

FILE\* file2;

if (((file1 = fopen(filename, "rb")) == NULL) || ((file2 = fopen(filename\_temp, "wb")) == NULL))

{

cout << "ЧЕТ НЕ ТО" << endl;

return;

}

int nums = n;

int count = 1;

float total = 0;

while (!feof(file1) && count <= nums)

{

Student student;

fread(&student, sizeof(Student), 1, file1);

total = 0;

for (int j = 0; j < 3; j++)

total += student.grades[j];

if (group == student.group && (total / 3 < average))

{

--n;

}

else fwrite(&student, sizeof(Student), 1, file2);

++count;

}

fclose(file1);

fclose(file2);

remove(filename);

rename(filename\_temp, filename);

return;

}

void addElemFile(Student new\_student) //добавление нового чел-ка после указанного

{

FILE\* file1;

FILE\* file2;

if (((file1 = fopen(filename, "rb")) == NULL) || ((file2 = fopen(filename\_temp, "wb")) == NULL))

{

cout << "ЧЕТ НЕ ТО" << endl;

return;

}

char after\_surname[255];

cout << "Введите фамилию, после которой хотите ввести пользователя: ";

cin.get();

cin.getline(after\_surname, 255);

int count = 1;

int num = n;

while (!feof(file1) && count <= num)

{

Student student;

fread(&student, sizeof(Student), 1, file1);

if (strcmp(after\_surname, student.surname) == 0 && (n - num == 0)) // (n - num == 0) - проверка на первое вхождение фамилии

{

fwrite(&student, sizeof(Student), 1, file2);

fwrite(&new\_student, sizeof(Student), 1, file2);

n++;

}

else

{

fwrite(&student, sizeof(Student), 1, file2);

}

++count;

}

fclose(file1);

fclose(file2);

remove(filename);

rename(filename\_temp, filename);

return;

}

int main()

{

setlocale(LC\_ALL, "Rus");

cout << "Введите кол-во обущающихся в группе: ";

cin >> n;

createFile(); //Создание файла;

outputFile(); //Вывод файла;

float average;

int group;

cout << endl << "Введите номер группы и среднее арифметическое оценок: ";

cin >> group >> average;

cout << endl;

deleteElemFile(group, average);

outputFile();

Student student; //дальше новый пользователь

{

cout << endl << "Введите нового пользователя: " << endl;

cout << "Введите ФИО: " << endl;

cin.get();

cout << "\* Введите фамилию: ";

cin.getline(student.surname, 255);

cout << "\* Введите имя: ";

cin.getline(student.name, 255);

cout << "\* Введите отчество: ";

cin.getline(student.surname2, 255);

cout << "Введите номер телефона: ";

cin >> student.telephone;

cout << "Введите номер группы: ";

cin.get();

cin >> student.group;

cout << "Введите 3 основных оценки через пробел: ";

for (int i = 1; i <= 3; i++) cin >> student.grades[i - 1];

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

addElemFile(student);

outputFile();

return 0;

}

Результат программы

