ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет прикладной математики, информатики и механики

Кафедра программного обеспечения и администрирования  
информационных систем

**Бинарные файлы**

Курсовая работа

02.03.03 – Математическое обеспечение и администрирование информационных систем

Зав. кафедрой д. ф.-м. н., проф. Артёмов М. А.

Обучающийся 1 курс, 9 группа Барбашина И. Н.

Руководитель д. ф.-м. н., проф. Артёмов M. A.

Воронеж 2023

# Аннотация

Данная работа позволяет выявить особенности при работе с бинарными и текстовыми файлами, а также узнать о возможностях взаимодействия с ними.

Тема данной работы имеет важное учебное значение и позволяет изучить принципы, идеи и возможности структурного программирования на примере выбранного языка программирования.

# Содержание

[Аннотация 2](#_Toc75897132)

[Содержание 3](#_Toc75897133)

[Введение 5](#_Toc75897134)

[1. Постановка задачи 6](#_Toc75897135)

[2. Анализ задачи 7](#_Toc75897136)

[2.1. Общий анализ 7](#_Toc75897137)

[2.2. Предполагаемые структуры данных 8](#_Toc75897138)

[2.3. Предполагаемая структура программы 9](#_Toc75897139)

[2.3.1. Предполагаемая структура модуля вспомогательных функций 9](#_Toc75897140)

2.3.2. Предполагаемая структура модуля для работы с одной записью….9

[2.3.3. Предполагаемая структура главной программы 10](#_Toc75897143)

[3. Средства реализации 12](#_Toc75897144)

[4. Требования к программному и аппаратному обеспечению 13](#_Toc75897145)

[5. Интерфейс пользователя 14](#_Toc75897146)

[6. Реализация 16](#_Toc75897147)

[6.1. Структуры данных 16](#_Toc75897148)

[6.2. Взаимодействие функциональных блоков 16](#_Toc75897149)

[6.3. Структура функциональных блоков 17](#_Toc75897150)

[6.4. Описание блоков 17](#_Toc75897151)

[6.4.1. Блок вспомогательных функций 17](#_Toc75897152)

[6.4.2. Блок функций для работы с файлом 17](#_Toc75897153)

[6.4.3. Блок функций для работы с записью 18](#_Toc75897154)

[6.4.4. Блок функций главной программы 18](#_Toc75897155)

6.4.5. Блок хранилище……………………………………………………...18

[7. Тестирование программы 19](#_Toc75897156)

[7.1. «Создание файла» 19](#_Toc75897157)

[7.2. «Добавление записей в файл» 19](#_Toc75897158)

[7.3. «Изменение записи в файле» 20](#_Toc75897159)

[7.4. «Удаление записи из файла» 20](#_Toc75897160)

[7.5. «Решение задачи» 20](#_Toc75897162)

[Заключение 21](#_Toc75897163)

[Список литературы 22](#_Toc75897164)

[Приложение. Листинг программы 23](#_Toc75897165)

# Введение

Данная курсовая работа посвящена изучению бинарных файлов и работе с ними в языке *C++.*

В языке *C++* файлом называется совокупность данных, записанная во внешней памяти и обладающая своим собственным именем и расширением.

Под бинарным же файлом понимается файл, информация в котором записана только в виде последовательности нулей и единиц. Работа с бинарными файлами обеспечивает прямой доступ к его записям, а также высокую скорость обмена информацией.

Цель данной работы – изучить возможности, предоставляемые языком *C++* для работы с бинарными файлами, и реализовать программу, которая будет обеспечивать выполнение заданных нам действий с бинарными файлами. Под этими действиями подразумевается создание, открытие, изменение (добавление и удаление) записей, конвертирование бинарного файла в текстовый, закрытие файла.

Тема данной работы имеет важное учебное значение и позволяет изучить принципы, идеи и возможности структурного программирования на примере выбранного языка программирования.

## Постановка задачи

Создайте файл «Студент». Подсчитайте средний балл каждой группы по заданному номеру курса и средний балл курса.

Также данная программа должна предоставлять пользователю основные инструменты для работы с файлами.

## Анализ задачи

### Общий анализ

Для решения поставленной нам задачи необходимо реализовать программу, предоставляющую нам базовые средства для работы с бинарными файлами и давать нам возможность их вызова по выбору пользователя.

Для этого нам необходимо разработать пакет инструментов, позволяющих выполнять нужные нам функции:

1. создавать и открывать бинарный файл;
2. отображать содержимое бинарного файла;
3. добавлять, изменять и удалять запись в данном файле;
4. конвертировать бинарный файл в текстовый;
5. сохранять файл;
6. выводить на консоль средние баллы групп заданного курса.

Также необходимо обеспечить контроль возможности совершения какого-либо действия с файлом.

Для решения частной задачи (расчет данных, в зависимости от введенного курса) необходимо осуществить просмотр всех имеющихся в файле данных и заносить числовые значения в специально для этого заведённый массив.

### Предполагаемые структуры данных

Для корректной обработки информации, хранящейся в бинарном файле, понадобятся вспомогательные структуры данных. Необходима структура, которая будет содержать данные одной записи бинарного файла.

Так как рабочий файл будет представлять из себя список информации о студентах, то запись должна иметь следующие поля:

1. ФИО студента – символьная строка длиной не более 40 символов;
2. курс студента – числовое значение не большее 4;
3. группа студента – числовое значение;
4. форма обучения – символьная строка длиной не более 20 символов;
5. результаты экзаменов – 4 поля вспомогательной структуры, отвечающей за результаты экзамена за один курс.

Структура, отвечающая за результаты экзамена за один курс, будет представлять из себя матрицу с 5 столбцами и 2 строками. Каждая строка отвечает за одну сессию на протяжение курса. Каждая клетка матрица является ещё одной вспомогательной структурой, отвечающей за результаты одного экзамена. Эта структура имеет только два поля: название экзамена – символьная строка длиной не более 40 символов, а также оценка за экзамен – числовое значение.

### Предполагаемая структура программы

Для удобства внутри программы выделим несколько модулей, каждый из которых будет содержать функции, выполняющие определённые группы действий.

#### Предполагаемая структура модуля вспомогательных функций

Один из модулей должен будет содержать вспомогательные функции, которые будут требоваться нам для взаимодействия между пользователем и программой.

1. Функция выбора числа из заданного интервала. Необходима для выбора какого-либо пункта меню, а также для ввода данных. Возвращает введённое из заданного интервала число.
2. Функция преобразования текстового файла в бинарный. Данная функция необходима для открытия файла.
3. Функция проверки файла. Данная функция необходима для понимания, есть ли ошибки в файле.
4. Функция чтения бинарного файла.
5. Функция-компаратор. Сортирует студентов в файле.

#### Предполагаемая структура модуля для работы с одной записью

Данный блок содержит функции, непосредственно работающие с файлом.

1. Функция печати файла. Данная функция должна выводить содержимое файла на экран.
2. Функция, которая осуществляет вывод информации о студенте на экран.
3. Функция изменения записи файла. Изменяет запись в файле.
4. Функция добавления записи в файл. Добавляет запись в заданный бинарный файл.
5. Функция удаления записи из файла. Удаляет выбранную запись из бинарного файла.
6. Функция, которая будет искать на каждом курсе средние баллы каждой группы.

#### Предполагаемая структура главной программы

Для удобства внутри проекта можно выделить несколько блоков, помимо основной программы, каждый из которых будет содержать подпрограммы, выполняющие определённые группы действий.

## Средства реализации

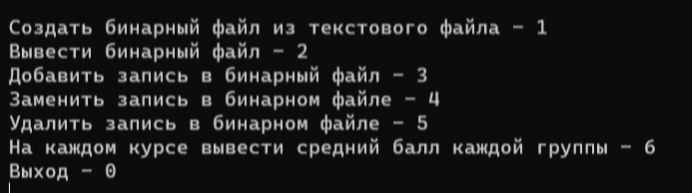
В качестве языка программирования был выбран язык *C++*. Возможности данного языка дают нам все необходимые инструменты для поставленных в данной работе целей и задач. При реализации данного проекта использовалась среда разработки *Microsoft Visual Studio 2022*.

## Требования к программному и аппаратному обеспечению

Приложение предназначено для использования на *IBM PC*-совместимых компьютерах с операционной системой *Windows XP* и новее.

Минимальные системные требования приложения соответствуют минимальным системным требованиям операционной системы.

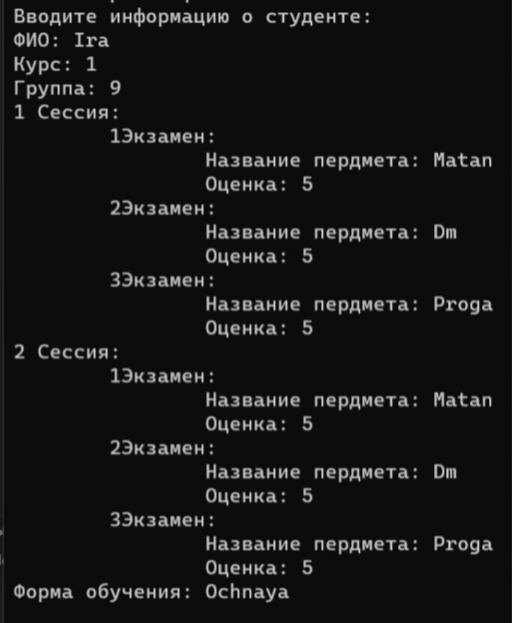
## Интерфейс пользователя

  
Рис. 5.1.

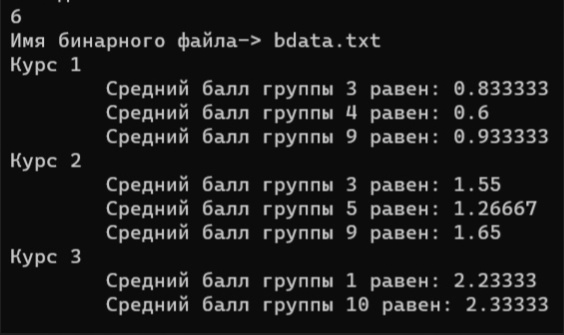
После создания или открытия (соответственно выбору пользователя) бинарного файла нам выводится меню, предоставляющее следующие действия на выбор:

1. создать бинарный файл;
2. вывести на консоль содержимое бинарного файла;
3. добавить запись;
4. заменить запись;
5. удалить запись;
6. решить требуемую задачу.

При выборе пункта «Добавить запись» пользователю предлагается ввести поля добавляемой записи. Пример работы функции рисунок 5.2.

  
Рис. 5.2.

При выборе пункта 6 происходит выполнение частной задачи, происходит выполнение частной задачи, пользователю предлагают ввести курс для получения результата. Пример работы рисунок 5.3.

  
Рис. 5.3.

## Реализация

### Структуры данных

Структура записей, из которых состоит файл:

struct Student

{

char FIO[SIZE\_STR];

int course;

int groupe;

Session sessions[CountSessions];

char form\_education[SIZE\_STR];

};

Вспомогательные структуры:

struct Exam

{

char title[SIZE\_STR];

int mark;

};

struct Session

{

Exam exams[CountExams];

};

### Взаимодействие функциональных блоков

Проект состоит из трех функциональных блоков. В каждом блоке расположены функции, отвечающие за определённый тип действий (взаимодействие с пользователем, обработка одной записи и взаимодействие с файлом).Схема взаимодействия между этими тремя блоками показана на рис. 6.1.

Главная программа

Блок хранилище

Блок для работы с файлами

Блок вспомогательных функций

Блок для работы с записью

Рис. 6.1. Схема взаимодействия блоков программы

### Структура функциональных блоков

В главной программе расположены подпрограммы, отвечающие за вывод меню на экран и доступность тех или иных пунктов меню. Из главной программы происходит вызов всех соответствующих процедур из других.

1. В функциональном блоке для работы с файлом расположены функции, предназначенные для работы непосредственно с файлом.
2. В функциональном блоке для работы с отдельной записью файла находятся функции, предназначенные для работы с одной записью.
3. В функциональном блоке вспомогательных функций расположены функции, обеспечивающие диалог между программой и пользователем, а также другие вспомогательные функции.

### Описание блоков

#### Блок вспомогательных функций

* f.read((char\*)&stud, sizeof(Student));

Данная функция используется для чтения бинарного файла.

* ftemp.write((char\*)&stud\_change, sizeof(Student));

Данная функция используется для записи в бинарный файл

* bool Compare(Student& s1, Student& s2);

#### Блок функций для работы с файлом

* void PrintFile(string file)

Вывод содержимого бинарного файла

* void AddRec()

Функция добавления студента

* void CnchRec()

Функция замены информации о студенте

* void DelRec()

Функция удаления информации о студенте

#### Блок функций для работы с записью

* bool ReadRec(Student& student)

Ввод информации о студенте с консоли

#### Блок функций главной программы

* void menu()

Функция вывода меню

* void main()

Главная функция, точка входа в программу

#### Блок хранилище

* void CreateFromText(string infile, string outfile)

Преобразование текстового файла в бинарный

* bool Compare(Student& s1, Student& s2)

Функция сортировки

## Тестирование программы

Для проверки корректности работы данной программы необходимо провести определенный набор тестов, каждый из которых будет проверять работоспособность определенной операции, производимой с файлом.

Тестирование данного приложения заключается в необходимости проверить функциональности всех подпрограмм, вызывающихся при работе приложения, и протестировать корректность обработки всех запросов пользователя (в том числе при некорректном вводе данных).

### «Создание файла»

*Цель теста:* проверить корректность создания файла с заданным именем.

*Описание теста и результат:* после запуска программы выбрать пункт «Создать бинарный файл из текстового файла» и ввести имя создаваемого файла. Если введённое имя некорректно, то должно будет быть выведено сообщение об неправильности введённого имени и просьба ввода имени заново. Если же файл с таким именем уже существует, то мы можем либо открыть существующий файл, либо вернуться в меню на уровень выше.

### «Добавление записей в файл»

*Цель теста:* проверить корректность добавления новых записей в файл.

*Описание теста и результат:* необходимо открыть или создать бинарный файл, а после этого, выбрав в меню пункт «Добавить запись в бинарный файл», ввести новые данные. После завершения можно выбрать пункт «Вывести бинарный файл», чтобы убедиться в корректности добавления записи.

### «Изменение записи в файле»

*Цель теста:* проверить корректность изменения информации в файле.

*Описание теста и результат:* необходимо открыть или создать бинарный файл, а после этого, выбрав в меню пункт «Заменить запись в бинарном файле», ввести номер изменяемой записи и ввести новые данные. В случае, если был введён неправильный номер записи, должно быть выведено сообщение об ошибке и просьба ввести новый номер. После завершения можно выбрать пункт «Вывести бинарный файл», чтобы убедиться в корректности изменения записи.

### «Удаление записи из файла»

*Цель теста:* проверить корректность удаления записей из файла.

*Описание теста и результат:* необходимо открыть или создать бинарный файл, а после этого, выбрав в меню пункт «Удалить запись в бинарном файле», ввести номер удаляемой записи. В случае, если был введён неправильный номер записи, должно быть выведено сообщение об ошибке и просьба ввести новый номер. В случае, если файл окажется пуст, нам будет выведено сообщение об этом. После завершения можно выбрать пункт «Вывести бинарный файл», чтобы убедиться в корректности удаления записи.

### «Решение задачи»

*Цель теста:* проверить правильность решения заданной задачи.

*Описание теста и результат:* необходимо открыть или создать бинарный файл. После выбора в меню пункта «На каждом курсе вывести средний балл каждой группы» нам в консоль будут выведены все эти баллы. В случае, если студентов на заданном курсе нет, нам будет выведено сообщение об этом.

# Заключение

Данная работа позволила изучить основные принципы и идеи структурного программирования на примере языка *C++*, а также выяснить особенности методов работы с бинарными файлами в этом языке.

В результате решения поставленной задачи была создана и протестирована программе, позволяющая выполнять следующие операции с бинарными файлами:

1. создавать и открывать бинарный файл;
2. отображать содержимое бинарного файла;
3. добавлять записи в бинарный файл;
4. изменять записи в бинарном файле;
5. удалять записи из бинарного файла;
6. вывести на консоль средние баллы всех групп на каждом курсе.

Управление программой происходит посредством меню и предоставляет пользователю возможность работать с бинарными файлами, в которых хранится информация в виде структур, используемых в этом проекте.

# Список литературы

Программирование на языке Паскаль: задачник / О. Ф. Ускова [и др.]; под ред. О. Ф. Усковой. — Санкт-Петербург: Питер, 2002. — 336 с.

Алгоритмы и структуры данных / Н. Вирт; пер. с англ. Л. Ю. Иоффе под ред. Д. Б. Подшивалова. — Москва: Мир, 1985. — 406 с.

Стиль, разработка, эффективность, отладка и испытание программ / Д. Ван Тассел; пер. с англ. Е. К. Масловского, В. А. Прониной; под ред. Э. А. Трахтенгерца. — Москва: Мир, 1981. — 319 с.

# Приложение. Листинг программы

//Барбашина Ирина, 9 группа, 1 курс, МОиАИС

//На каждом курсе вывести средние баллы каждой группы

#include <iostream>

#include <string>

#include <fstream>

#include <string.h>

#include <cstring>

using std::string;

using std::cout;

using std::cin;

const int CountMaxStudents = 35;

const int CountExams = 3;

const int CountSessions = 10;

const int SIZE\_STR = 70;

struct Exam

{

char title[SIZE\_STR];

int mark;

};

struct Session

{

Exam exams[CountExams];

};

struct Student

{

char FIO[SIZE\_STR];

int course;

int groupe;

Session sessions[CountSessions];

char form\_education[SIZE\_STR];

};

void menu();

void PrintFile(string file);

bool ReadRec(Student& student);

void CreateFromText(string infile, string outfile);

void CrtText();

void PrintFile();

void AddRec();

void CnchRec();

void DelRec();

bool Compare(Student& s1, Student& s2);

void SortMas(Student\* students, int& len, bool (\*cmp)(Student&, Student&));

void task();

string FileN;

int main()

{

system("chcp 65001");

setlocale(LC\_ALL, "Ru");

menu();

}

void menu()

{

int k = 0;

do

{

cout << "\n";

cout << "Создать бинарный файл из текстового файла - 1\n";

cout << "Вывести бинарный файл - 2\n";

cout << "Добавить запись в бинарный файл - 3\n";

cout << "Заменить запись в бинарном файле - 4\n";

cout << "Удалить запись в бинарном файле - 5\n";

cout << "На каждом курсе вывести средний балл каждой группы - 6\n";

cout << "Выход - 0\n";

cin >> k;

switch (k)

{

case 1:

CrtText();

break;

case 2:

PrintFile();

break;

case 3:

AddRec();

break;

case 4:

CnchRec();

break;

case 5:

DelRec();

break;

case 6:

task();

break;

}

} while (k != 0);

}

void PrintFile(string file)

//Вывод файла в консоль

{

std::fstream f(file.c\_str(), std::ios::binary | std::ios::in);

if (f.fail())

{

cout << "Не удалось открыть " + file + "\n";

return;

}

Student stud;

f.seekg(0);

int k = 1;

while (f.read((char\*)&stud, sizeof(Student)))

{

cout << "№ - " << k << "\n";

cout << "ФИО: " << stud.FIO << "\n";

cout << "Крус:" << stud.course << "\n";

cout << "Группа: " << stud.groupe << "\n";

for (int i = 0; i < stud.course \* 2; ++i)

{

cout << i + 1 << " Сессия:\n";

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

{

cout << '\t' << j + 1 << " Экзамен:\n";

cout << "\t\t" << "Название пердмета: ";

cout << stud.sessions[i].exams[j].title;

cout << "\n\t\tОценка: ";

cout << stud.sessions[i].exams[j].mark;

cout << '\n';

}

}

cout << "Форма обучения: " << stud.form\_education << "\n";

++k;

}

f.close();

}

bool ReadRec(Student& student)

//Считывание записи о студенте с консоли

{

char str[SIZE\_STR];

cout << "ФИО: ";

cin.getline(str, SIZE\_STR); // skip мусора

cin.getline(str, SIZE\_STR);

strcpy\_s(student.FIO, str);

cout << "Курс: ";

cin.getline(str, SIZE\_STR);

student.course = std::stoi(str);

cout << "Группа: ";

cin.getline(str, SIZE\_STR);

student.groupe = std::stoi(str);

for (int i = 0; i < 2 \* student.course; ++i)

{

cout << i + 1 << " Сессия:\n";

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

{

cout << "\t" << j + 1 << "Экзамен:\n";

cout << "\t\t" << "Название пердмета: ";

cin.getline(str, SIZE\_STR);

strcpy\_s(student.sessions[i].exams[j].title, str);

cout << "\t\t" << "Оценка: ";

cin.getline(str, SIZE\_STR);

student.sessions[i].exams[j].mark = std::stoi(str);

}

}

cout << "Форма обучения: ";

cin.getline(str, SIZE\_STR);

strcpy\_s(student.form\_education, str);

return true;

}

void CreateFromText(string infile, string outfile)

// Создание файла с текстового файла

{

std::fstream f(outfile.c\_str(), std::ios::binary | std::ios::out);

std::ifstream ft(infile.c\_str());

if (!ft)

{

cout << "Не открыт файл\n";

return;

}

string str = "";

Student students[CountMaxStudents];

int k = 0;

while (getline(ft, str))

{

strcpy\_s(students[k].FIO, str.c\_str());

std::getline(ft, str);

students[k].course = std::stoi(str);

std::getline(ft, str);

students[k].groupe = std::stoi(str);

for (int i = 0; i < students[k].course \* 2; ++i)

{

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

{

std::getline(ft, str);

strcpy\_s(students[k].sessions[i].exams[j].title, str.c\_str());

std::getline(ft, str);

students[k].sessions[i].exams[j].mark = std::stoi(str);

}

}

std::getline(ft, str);

strcpy\_s(students[k].form\_education, str.c\_str());

++k;

}

int len = k;

SortMas(students, len, Compare);

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

{

f.write((char\*)&students[i], sizeof(Student));

}

f.close();

cout << "Файл создан\n";

cout << "Запомнить имя бинарного файла? y/n\n";

char ch;

cin >> ch;

if (ch == 'y')

FileN = outfile;

else

FileN = "";

}

void CrtText()

{

string fileName;

cout << "Имя бинарного файла-> ";

cin >> fileName;

string strin;

cout << "Введите имя исходного файла-> ";

cin >> strin;

CreateFromText(strin, fileName);

}

void PrintFile()

{

string fileName;

if (FileN == "")

{

cout << "Имя бинарного файла-> ";

cin >> fileName;

}

else

fileName = FileN;

PrintFile(fileName);

}

void AddRec()

{

string fileName;

if (FileN == "")

{

cout << "Имя бинарного файла-> ";

cin >> fileName;

}

else

fileName = FileN;

std::fstream f(fileName.c\_str(), std::ios::binary | std::ios::in);

if (!f)

cout << "файла нет";

std::fstream ftemp("temp.data", std::ios::binary | std::ios::out);

Student stud, new\_stud;

cout << "Вводите информацию о студенте:\n";

bool flag = true;

if (ReadRec(new\_stud))

{

while (f.read((char\*)&stud, sizeof(Student)))

{

if (Compare(stud, new\_stud) && flag)

{

ftemp.write((char\*)&new\_stud, sizeof(Student));

flag = false;

}

ftemp.write((char\*)&stud, sizeof(Student));

}

if (flag)

{

ftemp.write((char\*)&new\_stud, sizeof(Student));

}

}

ftemp.close();

f.close();

remove(fileName.c\_str());

rename("temp.data", fileName.c\_str());

}

void CnchRec()

{

string fileName;

if (FileN == "")

{

cout << "Имя бинарного файла-> ";

cin >> fileName;

}

else

fileName = FileN;

std::fstream f(fileName.c\_str(), std::ios::binary | std::ios::in);

cout << "Номер заменяемой записи-> ";

int num;

cin >> num;

Student stud;

Student stud\_change;

std::fstream ftemp("temp.data", std::ios::binary | std::ios::out);

int k = 0;

bool flag = true;

ReadRec(stud\_change);

while (f.read((char\*)&stud, sizeof(Student)))

{

++k;

if (k != num)

{

if (Compare(stud, stud\_change) && flag)

{

ftemp.write((char\*)&stud\_change, sizeof(Student));

flag = false;

}

ftemp.write((char\*)&stud, sizeof(Student));

}

}

if (num > k)

cout << "Запись с таким номером не существует\n";

ftemp.close();

f.close();

remove(fileName.c\_str());

rename("temp.data", fileName.c\_str());

}

void DelRec()

// Удаление записи из файла

{

string fileName;

if (FileN == "")

{

cout << "Имя бинарного файла-> ";

cin >> fileName;

}

else

fileName = FileN;

std::fstream f(fileName.c\_str(), std::ios::binary | std::ios::in);

cout << "Номер удаляемой записи-> ";

int num;

cin >> num;

Student stud;

std::fstream ftemp("temp.data", std::ios::binary | std::ios::out);

int k = 0;

while (f.read((char\*)&stud, sizeof(Student)))

{

++k;

if (k != num)

ftemp.write((char\*)&stud, sizeof(Student));

}

if (num > k)

cout << "Запись с таким номером не существует\n";

ftemp.close();

f.close();

remove(fileName.c\_str());

rename("temp.data", fileName.c\_str());

}

bool Compare(Student& s1, Student& s2)

// Компаратор для сравнения студентов

{

string stud1\_FIO = s1.FIO;

string stud2\_FIO = s2.FIO;

if (s1.course > s2.course)

return true;

else

if (s1.course < s2.course)

return false;

else

if (s1.groupe > s2.groupe)

return true;

else

if (s1.groupe < s2.groupe)

return false;

else

if (stud1\_FIO >= stud2\_FIO)

return true;

else

return false;

}

void SortMas(struct Student arr[], int& len, bool (\*cmp)(Student&, Student&))

// Сортировка студентов

{

/\*for (int i = 0; i < len - 1; i++)

for (int j = 0; j < len - i - 1; j++)

if (cmp(students[j], students[j + 1]))

std::swap(students[j], students[j + 1]);\*/

for (int i = 1; i < len; i++)

{

Student key = arr[i];

int j = i - 1;

while (j >= 0 && cmp(arr[j], key))

{

arr[j + 1] = arr[j];

j--;

}

arr[j + 1] = key;

}

}

void task() {

// Вывести средний балл группы с каждого курса

string fileName;

if (FileN == "")

{

cout << "Имя бинарного файла-> ";

cin >> fileName;

}

else

fileName = FileN;

std::fstream f(fileName.c\_str(), std::ios::binary | std::ios::in);

Student stud;

double sred\_arifm = 0;

int sum = 0;

int group = 0;

int count = 0;

int course = 0;

int lastGroup = -1;

bool flag = true;

f.read((char\*)&stud, sizeof(Student));

while (flag)

{

course = stud.course;

cout << "Курс " << course << '\n';

do {

lastGroup = group;

group = stud.groupe;

if (lastGroup == group) {

flag = false;

break;

}

// посчитает средний балл

do

{

count++;

for (int i = 0; i < stud.course \* 2; ++i)

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

sum += stud.sessions[i].exams[j].mark;

} while (f.read((char\*)&stud, sizeof(Student)) && stud.course == course && stud.groupe == group);

sred\_arifm = sum / (double)(count \* CountExams \* CountSessions);

cout << "\tСредний балл группы " << group << " равен: " << sred\_arifm << '\n';

sum = 0;

count = 0;

} while (stud.course == course);

}

f.close();

}