Федеральное государственное автономное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет», ПНИПУ

ЛАБОРАТОРНАЯ РАБОТА

БЛОКОВЫЙ ВВОД-ВЫВОД

Выполнил: студент группы РИС-23-3б

Артем Владимирович Швецов

Проверила: доцент кафедры ИТАС

Ольга Андреевна Полякова

Пермь 2024

**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;
* 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**

* Структуры формируются специальным пользовательским типом с применением struct
* Функции на вход получают указатель на дополнительный буферный массив
* Внутри функций происходит обращение к файлу, с которым ведется работа
* При каждом открытии файла необходимо его закрытие для корректной работы.

**Блок-схема**

В конце документа

**Код**

#include <iostream>

#include <fstream>

#include <string>

#include <stdio.h>

using namespace std;

struct student

{

string sname;

string nname;

string fname;

string phone\_number;

string group;

int grade\_1;

int grade\_2;

int grade\_3;

void info\_input()

{

cout << "Введите фамилию студента: ";

cin >> sname;

cout << "Введите имя студента: ";

cin >> nname;

cout << "Введите отчество студента: ";

cin >> fname;

cout << "Введите номер телефона студента: ";

cin >> phone\_number;

cout << "Введите группу студента: ";

cin >> group;

cout << "Введите оценку студента за 1-ый предмет: ";

cin >> grade\_1;

cout << "Введите оценку студента за 2-ой предмет: ";

cin >> grade\_2;

cout << "Введите оценку студента за 3-ий предмет: ";

cin >> grade\_3;

}

void info\_output()

{

cout << "=================================================================================================" << endl << endl;

cout << "Фамилия студента: " << sname << endl;

cout << "Имя студента: " << nname << endl;

cout << "Отчество студента: " << fname << endl;

cout << "Номер телефона студента: " << phone\_number << endl;

cout << "Группа студента: " << group << endl;

cout << "Оценка студента за 1-ый предмет: " << grade\_1 << endl;

cout << "Оценка студента за 2-ой предмет: " << grade\_2 << endl;

cout << "Оценка студента за 3-ий предмет: " << grade\_3 << endl;

cout << endl << "=================================================================================================" << endl;

}

bool for\_deliting(string p\_num, double mid\_gr) {

if ((phone\_number == p\_num) && (((double)(grade\_1 + grade\_2 + grade\_3) / 3.0) < mid\_gr)) {

return true;

}

else {

return false;

}

}

};

void fun\_form(int n, student\* stud);

void fun\_print(int n, student\* mas);

int fun\_delete\_matchng(int n, string p\_num, double mid\_gr, student\* mas);

void fun\_add(int n, string name, student\* mas);

int main()

{

system("chcp 1251");

int n;

string p\_num, sname;

int mid\_gr;

cout << "Введите количество студентов: ";

cin >> n;

student\* studs = new student[n];

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

{

studs[i].info\_input();

}

student mas[50];

fun\_form(n, studs);

fun\_print(n, mas);

cout << "Введите номер на удаление: ";

cin >> p\_num;

cout << "Введите среднюю оценку на удаление: ";

cin >> mid\_gr;

n = fun\_delete\_matchng(n, (string)p\_num, mid\_gr, mas);

fun\_print(n, mas);

cout << "Введите фамилию, после которой нужно добавить элемент: ";

cin >> sname;

fun\_add(n, sname, mas);

n++;

fun\_print(n, mas);

return 0;

}

void fun\_form(int n, student\* w\_mas)

{

FILE\* f;

if ((f = fopen("database\_students.dat", "wb")) == NULL) exit(1);

student\* stud = w\_mas;

fwrite(stud, sizeof(student), n, f);

fclose(f);

}

void fun\_print(int n, student\* mas)

{

FILE\* f;

if ((f = fopen("database\_students.dat", "rb")) == NULL) exit(2);

student\* stud = mas;

fread(stud, sizeof(student), n, f);

fclose(f);

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

{

stud[i].info\_output();

}

}

int fun\_delete\_matchng(int n, string p\_num, double mid\_gr, student\* mas)

{

int dels = 0;

student\* stud = mas;

FILE\* f;

if ((f = fopen("database\_students.dat", "rb")) == NULL) exit(3);

fread(stud, sizeof(student), n, f);

fclose(f);

int new\_n = n;

for (int i = 0; i < new\_n; ++i) {

if (stud[i].for\_deliting(p\_num, mid\_gr))

{

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

{

stud[j] = stud[j + 1];

}

dels++;

new\_n--;

}

}

cout << dels << endl;

if ((f = fopen("database\_students.dat", "wb")) == NULL) exit(4);

fwrite(stud, sizeof(student), n, f);

fclose(f);

return new\_n;

}

void fun\_add(int n, string name, student\* mas)

{

student\* stud = mas;

FILE\* f;

if ((f = fopen("database\_students.dat", "rb")) == NULL) exit(3);

fread(stud, sizeof(student), n + 1, f);

fclose(f);

int i = 0;

bool check = true;

while ((i < n) && (check)) {

if (stud[i].sname == name)

{

for (int j = n + 1; j > i + 1; j--)

{

stud[j] = stud[j - 1];

}

check = false;

--i;

}

++i;

}

if (check) {

cout << "Ошибка поиска";

}

else

{

stud[i + 1].info\_input();

}

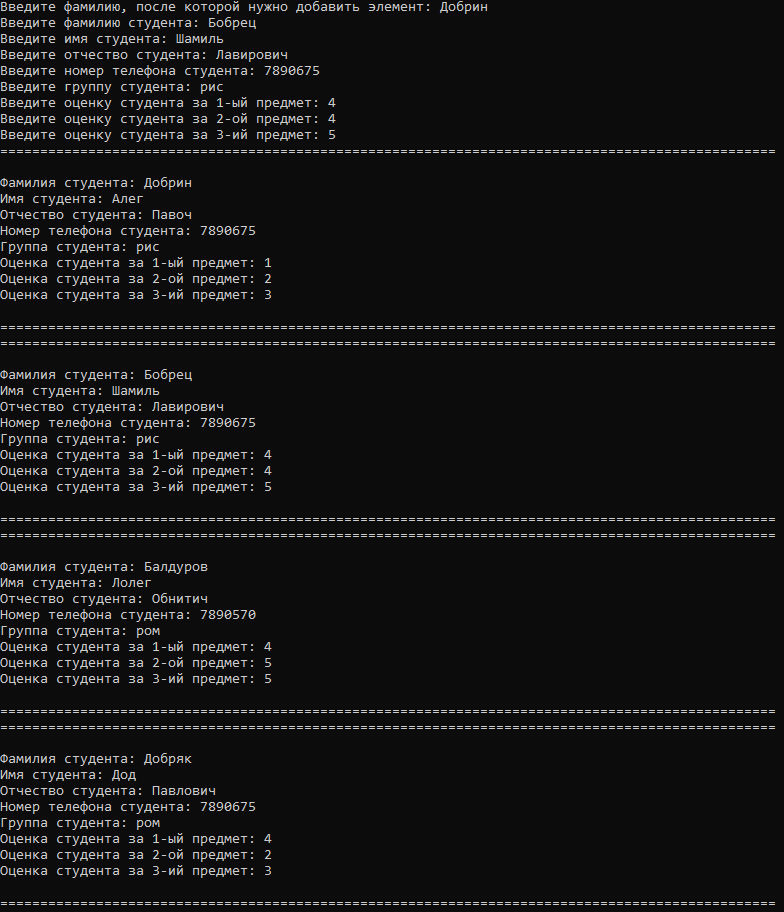
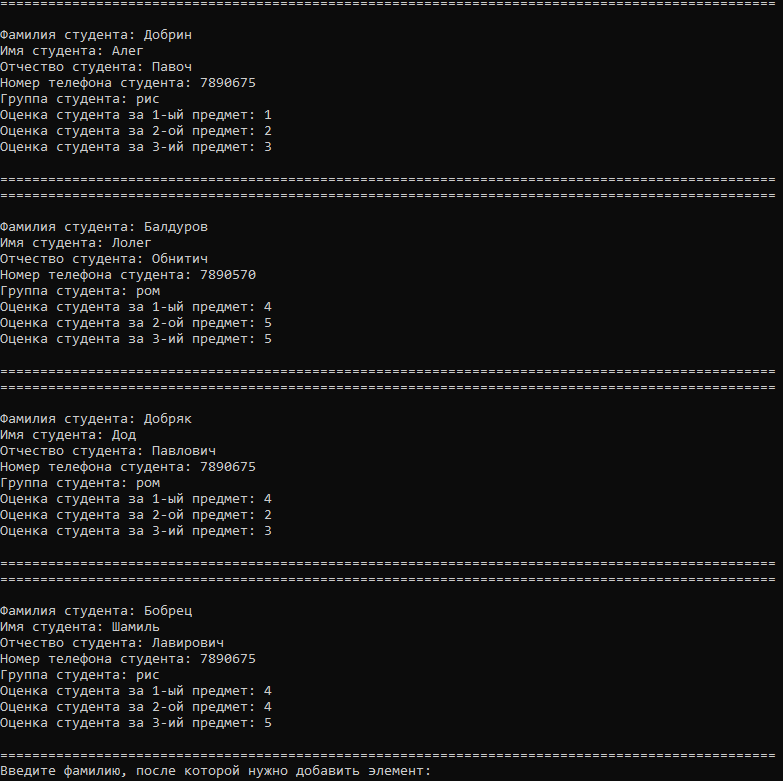
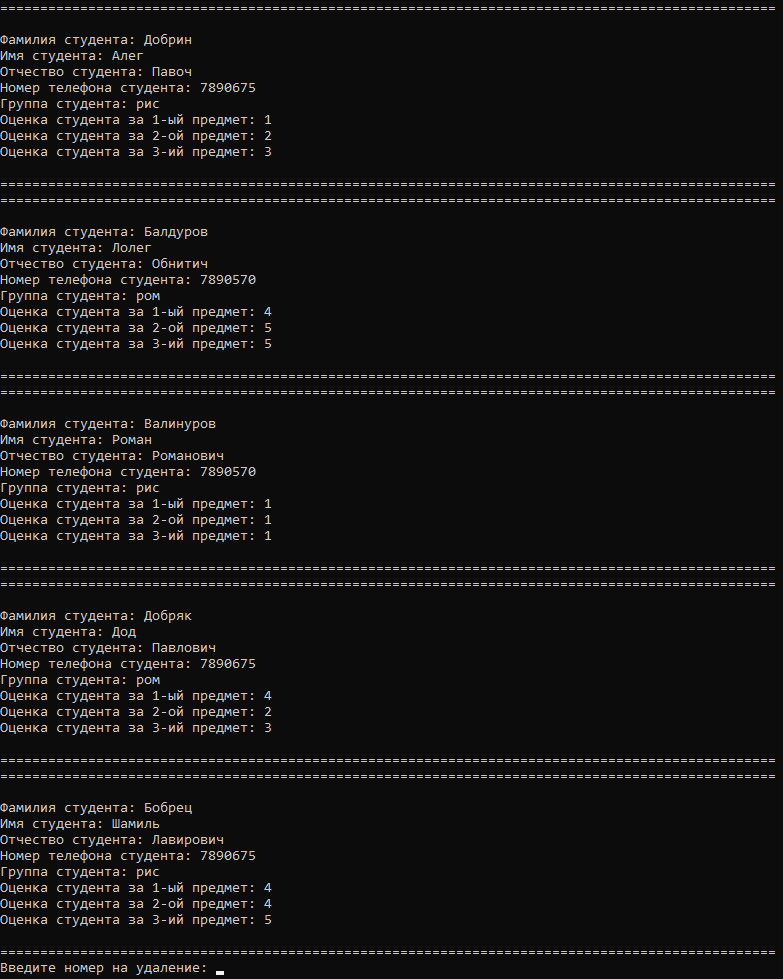
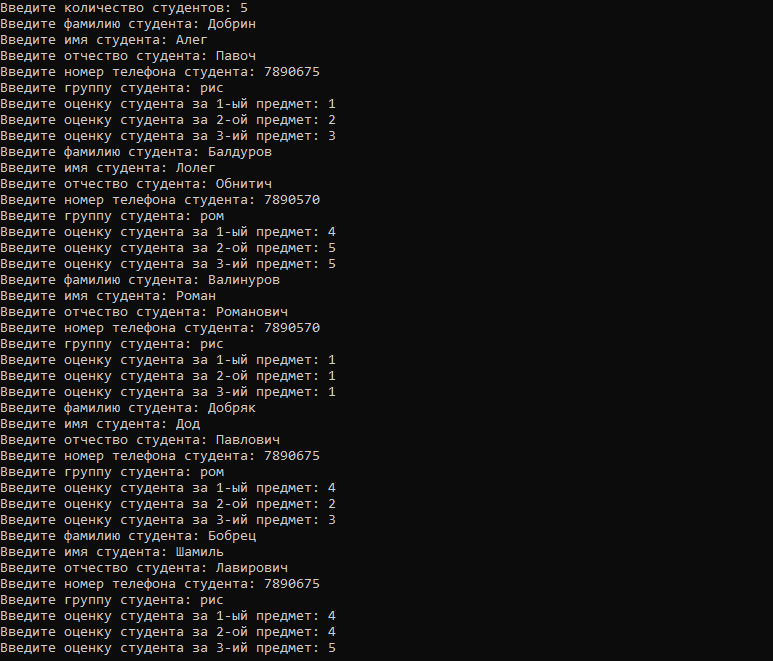
if ((f = fopen("database\_students.dat", "wb")) == NULL) exit(4);

fwrite(stud, sizeof(student), n + 1, f);

fclose(f);

}

**Решение**



**Выводы**

В ходе данной работы были изучены структуры и блоковый ввод-вывлд и некоторые особенности работы с ними.

**Github**

<https://github.com/Hitikov/Lab_8>

