БГУИР

Кафедра ВМиП

Отчет по лабораторной работе № 7

Тема: «Обработка структур с использованием файлов»

Вариант 13

Выполнил:

студент группы 324402 Цевелюк А.И.

Проверила:

ассистент каф. ВМиП Новицкая Л.И.

Минск

2023

Цель работы

Цель работы: изучить правила создания и обработки данных структурного типа с использованием файлов, написать и отладить программу по созданию файлов.

Задание

Задание: написать программу обработки файла записей, содержащую следующие пункты меню: «Создание», «Просмотр», «Добавление», «Редактирование», «Удаление», «Сортировка», «Решение индивидуального задания».

Каждая запись должны содержать следующую информацию о студенте: фамилия, номер группы, оценки за семестр по физике, математике и информатике, средний балл.

Организовать ввод входных данных, средний балл рассчитать по введенным оценкам.

Индивидуальное задание:

13. Вычислить общий средний балл студентов интересующей вас группы и распечатать список студентов этой группы, имеющих средний балл выше общего.

Листинг программы

#include <iostream>

#include <fstream>

#include <string>

#include <iomanip>

#include <algorithm>

#include <vector>

using namespace std;

struct student

{

unsigned id;

string surname;

unsigned group\_number;

unsigned physics;

unsigned math;

unsigned inf;

double average;

};

bool compare\_by\_id(const student& student1, const student& student2)

{

return student1.id < student2.id;

}

bool compare\_by\_surname(const student& student1, const student& student2)

{

return student1.surname < student2.surname;

}

bool compare\_by\_average(const student& student1, const student& student2)

{

return student1.average < student2.average;

}

bool compare\_by\_group\_number(const student& student1, const student& student2)

{

return student1.group\_number < student2.group\_number;

}

void read\_file(unsigned& students\_count, vector<student>& students\_array)

{

ifstream file("students.txt");

string buffer;

getline(file, buffer);

students\_count = stoi(buffer);

for (int i = 0; i < students\_count; i++)

{

student push\_back\_student;

getline(file, buffer, ' ');

push\_back\_student.id = stoi(buffer);

getline(file, buffer, ' ');

push\_back\_student.surname = buffer;

getline(file, buffer, ' ');

push\_back\_student.group\_number = stoi(buffer);

getline(file, buffer, ' ');

push\_back\_student.physics = stoi(buffer);

getline(file, buffer, ' ');

push\_back\_student.math = stoi(buffer);

getline(file, buffer, ' ');

push\_back\_student.inf = stoi(buffer);

getline(file, buffer, '\n');

push\_back\_student.average = stod(buffer);

students\_array.push\_back(push\_back\_student);

}

file.close();

}

void rewrite\_file(const unsigned& students\_count, const vector<student>& students\_array)

{

ofstream file("temp.txt");

file << students\_count << '\n';

for (int i = 0; i < students\_count; i++)

{

file << students\_array[i].id << ' ';

file << students\_array[i].surname << ' ';

file << students\_array[i].group\_number << ' ';

file << students\_array[i].physics << ' ';

file << students\_array[i].math << ' ';

file << students\_array[i].inf << ' ';

file << students\_array[i].average << '\n';

}

file.close();

remove("students.txt");

rename("temp.txt", "students.txt");

}

void create\_new\_file(unsigned& students\_count, vector<student>& students\_array)

{

ofstream file("temp.txt");

student test\_student{ 1, "Ivanov", 324401, 1, 1, 1, 1 };

students\_count = 1;

students\_array.clear();

students\_array.push\_back(test\_student);

file << students\_count << '\n' << test\_student.id << " " << test\_student.surname << " " << test\_student.group\_number << " ";

file << test\_student.physics << " " << test\_student.math << " " << test\_student.inf << " " << test\_student.average << '\n';

file.close();

remove("students.txt");

rename("temp.txt", "students.txt");

}

void print\_file(const unsigned& students\_count, const vector<student>& students\_array)

{

system("cls");

cout << "--------------------------------StudentsDatabase v. 0.0.1--------------------------------\n\n";

cout << "ID:" << setw(12) << "Surname" << setw(20) << "Group number:" << setw(12) << "Physics:" << setw(12) << "Math:" << setw(12) << "Inf:" << setw(20) << "Average:\n\n";

for (int i = 0; i < students\_count; i++)

{

cout << students\_array[i].id << setw(15) << students\_array[i].surname << setw(18) << students\_array[i].group\_number << setw(13);

cout << students\_array[i].physics << setw(13) << students\_array[i].math << setw(12) << students\_array[i].inf << setw(15) << students\_array[i].average << '\n';

}

cout << "\n-----------------------------------------------------------------------------------------\n";

}

void add\_student(unsigned& students\_count, vector<student>& students\_array)

{

string local\_choice;

do

{

system("cls");

students\_count++;

student student\_to\_add;

student\_to\_add.id = students\_count;

cout << "Enter surname: ";

cin >> student\_to\_add.surname;

cout << "\nEnter group number: ";

cin >> student\_to\_add.group\_number;

cout << "\nEnter physics mark: ";

cin >> student\_to\_add.physics;

cout << "\nEnter maths mark: ";

cin >> student\_to\_add.math;

cout << "\nEnter informatics mark: ";

cin >> student\_to\_add.inf;

student\_to\_add.average = (student\_to\_add.physics + student\_to\_add.math + student\_to\_add.inf) / 3.;

students\_array.push\_back(student\_to\_add);

cout << "\nDo you want to add one more student (y/n)? ";

cin >> local\_choice;

} while (local\_choice != "n");

rewrite\_file(students\_count, students\_array);

}

void delete\_student(unsigned& students\_count, vector<student>& students\_array)

{

unsigned student\_to\_delete\_id, student\_to\_delete\_number;

cout << "\nEnter student to delete ID: ";

cin >> student\_to\_delete\_id;

for (int i = 0; i < students\_count; i++)

{

if (students\_array[i].id == student\_to\_delete\_id)

{

student\_to\_delete\_number = i;

break;

}

}

students\_count--;

if (students\_count == 0)

create\_new\_file(students\_count, students\_array);

else

{

students\_array.erase(students\_array.begin() + student\_to\_delete\_number);

for (int i = 0; i < students\_count; i++)

{

if (students\_array[i].id > student\_to\_delete\_id)

students\_array[i].id--;

}

rewrite\_file(students\_count, students\_array);

}

}

void edit\_student(const unsigned& students\_count, vector<student>& students\_array)

{

unsigned student\_to\_edit\_id;

cout << "\nEnter student to edit ID: ";

cin >> student\_to\_edit\_id;

for (int i = 0; i < students\_count; i++)

{

if (students\_array[i].id = student\_to\_edit\_id)

{

student\_to\_edit\_id = i;

break;

}

}

cout << "\nEnter surname: ";

cin >> students\_array[student\_to\_edit\_id].surname;

cout << "\nEnter group number: ";

cin >> students\_array[student\_to\_edit\_id].group\_number;

cout << "\nEnter physics mark: ";

cin >> students\_array[student\_to\_edit\_id].physics;

cout << "\nEnter maths mark: ";

cin >> students\_array[student\_to\_edit\_id].math;

cout << "\nEnter informatics mark: ";

cin >> students\_array[student\_to\_edit\_id].inf;

students\_array[student\_to\_edit\_id].average = (students\_array[student\_to\_edit\_id].physics + students\_array[student\_to\_edit\_id].math + students\_array[student\_to\_edit\_id].inf) / 3.;

rewrite\_file(students\_count, students\_array);

}

void sort\_students(const unsigned& students\_count, vector<student>& students\_array)

{

unsigned sort\_type;

cout << "\nHow do you want to sort students? (by surname - 1, by group number - 2, by average mark - 3, by ID - 4): ";

cin >> sort\_type;

switch (sort\_type)

{

case 1:

sort(students\_array.begin(), students\_array.end(), compare\_by\_surname);

break;

case 2:

sort(students\_array.begin(), students\_array.end(), compare\_by\_group\_number);

break;

case 3:

sort(students\_array.begin(), students\_array.end(), compare\_by\_average);

break;

case 4:

sort(students\_array.begin(), students\_array.end(), compare\_by\_id);

break;

default:

break;

}

rewrite\_file(students\_count, students\_array);

}

void personal\_task(const unsigned& students\_count, const vector<student>& students\_array)

{

unsigned group\_number;

double average = 0;

vector<student> group;

cout << "\nEnter group number: ";

cin >> group\_number;

for (int i = 0; i < students\_count; i++)

{

if (students\_array[i].group\_number == group\_number)

{

group.push\_back(students\_array[i]);

average += students\_array[i].average;

}

}

average = average / group.size();

cout << "\nAverage mark: " << average;

cout << "\nStudents higher than average: ";

for (int i = 0; i < group.size(); i++)

{

if (group[i].average > average)

cout << group[i].surname << " ";

}

cout << "\n";

system("pause");

}

void navigation(unsigned& students\_count, vector<student>& students\_array, unsigned& choice)

{

cout << "Add new student - 1\n";

cout << "Delete existing student - 2\n";

cout << "Sort students - 3\n";

cout << "Edit student - 4\n";

cout << "Personal task - 5\n";

cout << "Exit - any other\n";

cin >> choice;

switch (choice)

{

case 1:

add\_student(students\_count, students\_array);

break;

case 2:

delete\_student(students\_count, students\_array);

break;

case 3:

sort\_students(students\_count, students\_array);

break;

case 4:

edit\_student(students\_count, students\_array);

break;

case 5:

personal\_task(students\_count, students\_array);

break;

default:

system("cls");

cout << "Have a good day!" << '\n';

break;

}

}

int main()

{

vector<student> students\_array;

unsigned students\_count, choice = 0;

cout << "Work with current file (1) or create new one (2)? ";

cin >> choice;

if (choice == 1)

{

read\_file(students\_count, students\_array);

do

{

print\_file(students\_count, students\_array);

navigation(students\_count, students\_array, choice);

} while (choice < 6);

}

if (choice == 2)

{

create\_new\_file(students\_count, students\_array);

read\_file(students\_count, students\_array);

do

{

print\_file(students\_count, students\_array);

navigation(students\_count, students\_array, choice);

} while (choice < 6);

}

return 0;

}

Результат работы программы

