**Отчет по 4 лабораторной работе**

**Олейников Артём, группа НБИ-БД-23-02**

**Задание:**

Написать простую СУБД в которая будет удовлетворять следующим критериям:

Запись / чтение файла.

Загрузка из файла в оперативную память.

Добавление/редактирование/удаление строк(записей).

Поиск и вывод записи на экран.

Будет единственная таблица. Таблица должна содержать не менее 5 столбцов.

Предметная область на выбор студента например: Реестр машин, афиша кинотеатра, карточка товара в интернет магазине и т.д.

Интерфейс коммандная строка. Возможность выполнять несколько операций последовательно без перезапуска программы.

**Код:**

#include <fstream>

#include <iostream>

#include <sstream>

#include <string>

using namespace std;

struct Movie {

string titleFilm; // Название фильма

string sessionTime; // Время сеанса

string duration; // Продолжительность фильма

string hallNumber; // Номер зала

string genre; // Жанр фильма

string ageRestriction; // Возрастное ограничение

};

const int MAX\_MOVIES = 1000; // Максимальное количество записей

Movie movies[MAX\_MOVIES]; // Массив для хранения информации о фильмах

int movieCount = 0; // Текущее количество фильмов в базе данных

void loadFromFile(const string &filename) {

ifstream file(filename);

if (!file.is\_open()) {

cerr << "Ошибка при открытии файла" << endl;

return;

}

movieCount = 0;

string line;

while (getline(file, line)) {

if (movieCount >= MAX\_MOVIES)

break;

stringstream ss(line);

getline(ss, movies[movieCount].titleFilm, ',');

getline(ss, movies[movieCount].sessionTime, ',');

getline(ss, movies[movieCount].duration, ',');

getline(ss, movies[movieCount].hallNumber, ',');

getline(ss, movies[movieCount].genre, ',');

getline(ss, movies[movieCount].ageRestriction, ',');

movieCount++;

}

file.close();

}

void saveToFile(const string &filename) {

ofstream file(filename);

if (!file.is\_open()) {

cerr << "Ошибка при открытии файла" << endl;

return;

}

for (int i = 0; i < movieCount; i++) {

file << movies[i].titleFilm << "," << movies[i].sessionTime << ","

<< movies[i].duration << "," << movies[i].hallNumber << ","

<< movies[i].genre << "," << movies[i].ageRestriction << endl;

}

file.close();

}

void addMovie() {

if (movieCount >= MAX\_MOVIES) {

cerr << "Невозможно добавить больше фильмов" << endl;

return;

}

cin.ignore(); // Очистить буфер ввода

cout << "Введите название: ";

getline(cin, movies[movieCount].titleFilm);

cout << "Введите время сеанса (чч:мм): ";

getline(cin, movies[movieCount].sessionTime);

cout << "Введите продолжительность в часах: ";

getline(cin, movies[movieCount].duration);

cout << "Введите номер зала: ";

getline(cin, movies[movieCount].hallNumber);

cout << "Введите жанр: ";

getline(cin, movies[movieCount].genre);

cout << "Введите возрастное ограничение: ";

getline(cin, movies[movieCount].ageRestriction);

movieCount++;

}

void editMovie() {

cin.ignore(); // Очистить буфер ввода

string title;

cout << "Введите название фильма для редактирования: ";

getline(cin, title);

for (int i = 0; i < movieCount; i++) {

if (movies[i].titleFilm == title) {

cout << "Редактирование фильма с названием: " << title << endl;

while (true) {

cout << "Выберите атрибут для редактирования:" << endl;

cout << "1. Время сеанса" << endl;

cout << "2. Продолжительность" << endl;

cout << "3. Номер зала" << endl;

cout << "4. Жанр" << endl;

cout << "5. Возрастное ограничение" << endl;

cout << "6. Вернуться в главное меню" << endl;

cout << "Введите выбор: ";

int choice;

cin >> choice;

cin.ignore(); // Очистить буфер ввода после ввода числа

switch (choice) {

case 1:

cout << "Введите новое время сеанса: ";

getline(cin, movies[i].sessionTime);

break;

case 2:

cout << "Введите новую продолжительность: ";

getline(cin, movies[i].duration);

break;

case 3:

cout << "Введите новый номер зала: ";

getline(cin, movies[i].hallNumber);

break;

case 4:

cout << "Введите новый жанр: ";

getline(cin, movies[i].genre);

break;

case 5:

cout << "Введите новое возрастное ограничение: ";

getline(cin, movies[i].ageRestriction);

break;

case 6:

return;

default:

cout << "Неправильный выбор. Попробуйте еще раз." << endl;

}

}

}

}

cout << "Фильм с названием " << title << " не найден." << endl;

}

void deleteMovie() {

cin.ignore(); // Очистить буфер ввода

string title;

cout << "Введите название фильма для удаления: ";

getline(cin, title);

for (int i = 0; i < movieCount; i++) {

if (movies[i].titleFilm == title) {

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

movies[j] = movies[j + 1];

}

movieCount--;

cout << "Фильм с названием " << title << " удален." << endl;

return;

}

}

cout << "Фильм с названием " << title << " не найден." << endl;

}

void searchMovie() {

cin.ignore(); // Очистить буфер ввода

string title;

cout << "Введите название фильма для поиска: ";

getline(cin, title);

for (int i = 0; i < movieCount; i++) {

if (movies[i].titleFilm == title) {

cout << "Фильм найден:" << endl;

cout << "Название: " << movies[i].titleFilm << endl;

cout << "Время сеанса: " << movies[i].sessionTime << endl;

cout << "Продолжительность: " << movies[i].duration << endl;

cout << "Номер зала: " << movies[i].hallNumber << endl;

cout << "Жанр: " << movies[i].genre << endl;

cout << "Возрастное ограничение: " << movies[i].ageRestriction << endl;

return;

}

}

cout << "Фильм с названием " << title << " не найден." << endl;

}

void displayAllMovies() {

if (movieCount == 0) {

cout << "Нет фильмов в базе данных." << endl;

return;

}

for (int i = 0; i < movieCount; i++) {

cout << "Название: " << movies[i].titleFilm << endl;

cout << "Время сеанса: " << movies[i].sessionTime << endl;

cout << "Продолжительность: " << movies[i].duration << endl;

cout << "Номер зала: " << movies[i].hallNumber << endl;

cout << "Жанр: " << movies[i].genre << endl;

cout << "Возрастное ограничение: " << movies[i].ageRestriction << endl;

cout << endl;

}

}

int main() {

string filename = "movies.txt";

loadFromFile(filename);

while (true) {

cout << "1. Добавить фильм" << endl;

cout << "2. Редактировать фильм" << endl;

cout << "3. Удалить фильм" << endl;

cout << "4. Найти фильм" << endl;

cout << "5. Показать фильмы" << endl;

cout << "6. Сохранить и выйти" << endl;

cout << "Номер действия: ";

int choice;

cin >> choice;

switch (choice) {

case 1:

addMovie();

break;

case 2:

editMovie();

break;

case 3:

deleteMovie();

break;

case 4:

searchMovie();

break;

case 5:

displayAllMovies();

break;

case 6:

saveToFile(filename);

return 0;

default:

cout << "Неправильный выбор. Попробуйте еще раз." << endl;

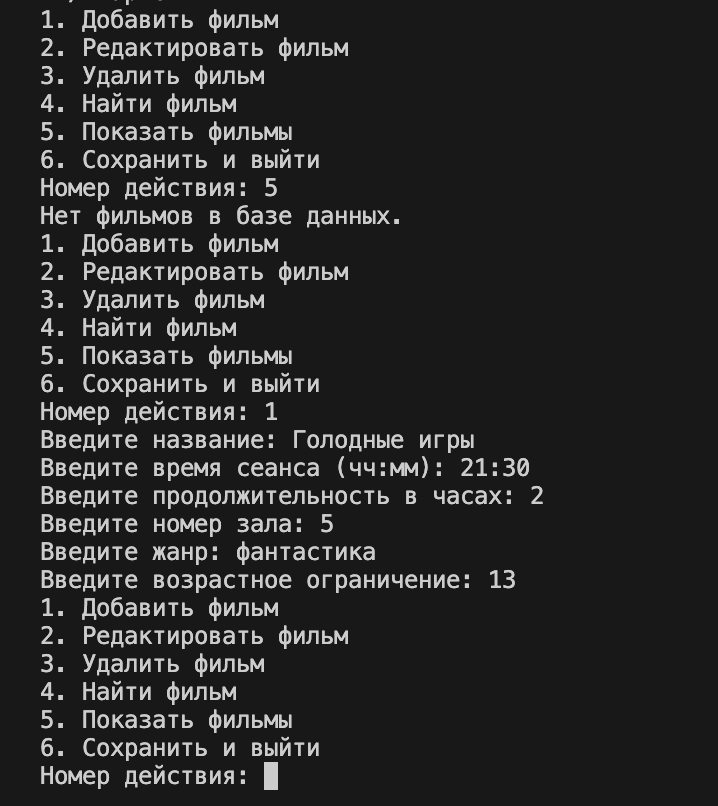
}

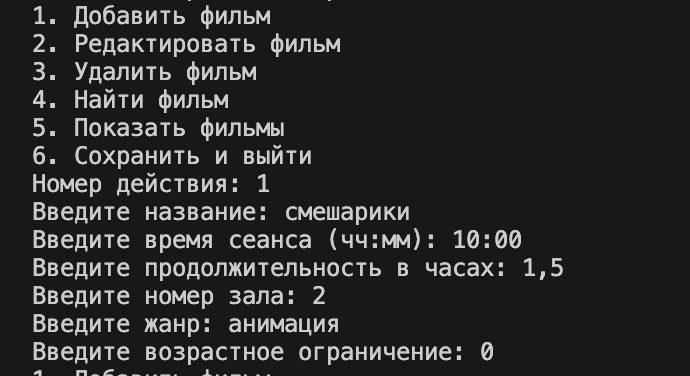
}

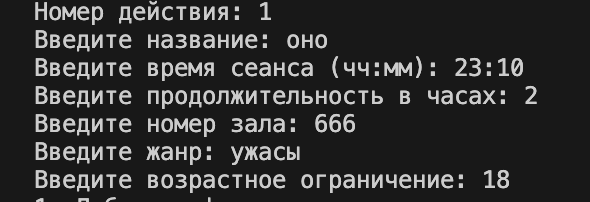
return 0;

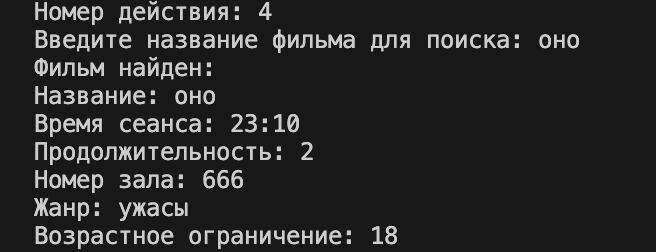
}

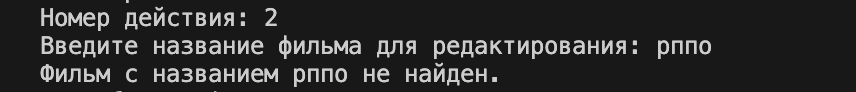
**Проверки:**

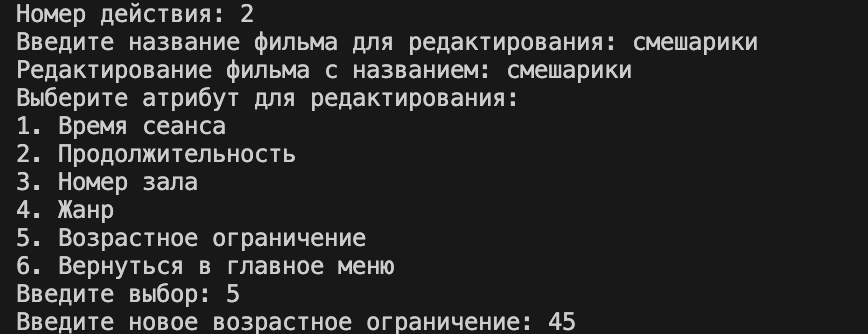
****

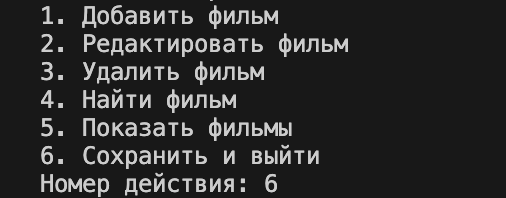
****

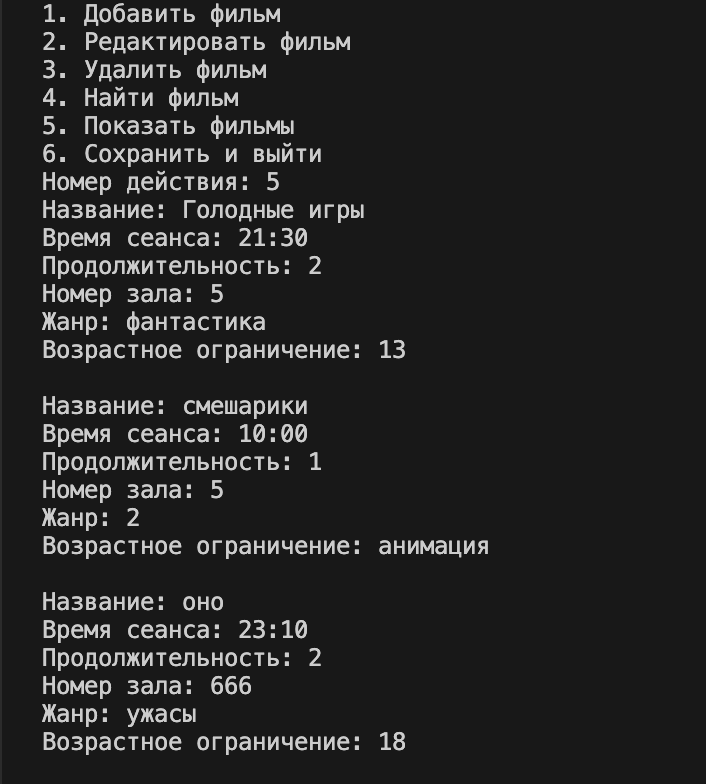
****

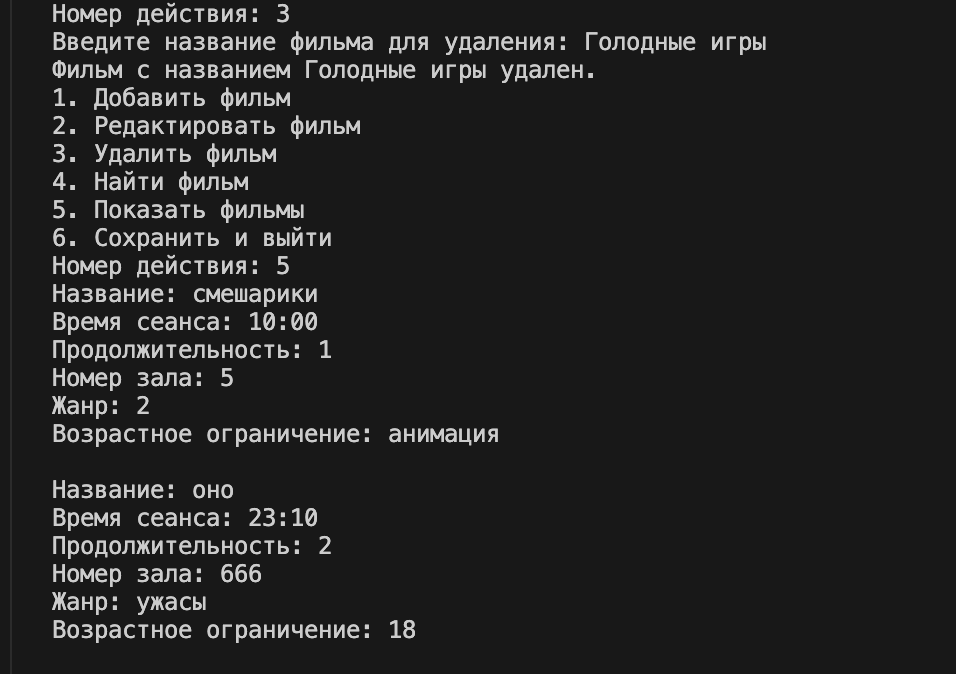
****

****

****

****

****

****