

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Лабораторная работа №2

по дисциплине: «ООП»

Выполнил: ст. группы ПВ-211

Чувилко Илья Романович

Проверил:

Буханов Дмитрий Геннадьевич

Харитонов Сергей Дмитриевич

Белгород 2023 г.

Тема: Модульное программирование. Интерфейсы.

Цель работы: Получение навыков модульной декомпозиции предметной области, создания модулей. Разработка интерфейсов.

Задание: Разработать программу, состоящую из трех модулей в соответствии с указанным вариантом задания. Первый модуль – основной код программы; второй содержит интерфейсы; третий модуль – реализацию этих интерфейсов. Количество структур данных ("объектов") не менее пяти.

Содержание отчета:

1. Тема, цель работы, вариант задания.
2. Реализация задачи на языке C++.

Основной код программы:

```
#include "FilmLibrary.h"

int main() {
    vector<Movie> m = {
        {"Первый", 10, {"a", "b"}},
        {"Второй", 23, {"b", "c"}},
        {"Третий", 1, {"a", "c"}},
        {"Четвертый", 2, {"c", "d"}}
    };
    Explorer::movies = m;

    while (true) {
        Explorer::outputMovies();
        cout << "Введите:\n "
            "ft - поиск фильма по названию\n"
            "fg - поиск фильма по вашему любимому жанру\n"
            "play - воспроизведение фильма по названию\n"
            "u - загрузка своего фильма\n"
            "q - выход из программы\n"
            "Так что же вы собираетесь делать? ";
        string input;
        cin >> input;
        if (input == "ft")
            Finder::findByTitle();
        else if (input == "fg")
            Finder::findByGenres();
        else if (input == "play")
            Player::play();
        else if (input == "u")
            Upload::upload();
        else if (input == "q")
            break;
    }

    return 0;
}
```

Содержимое заголовочного файла:

```
#ifndef CODE_FILMLIBRARY_H
#define CODE_FILMLIBRARY_H

#include <iostream>
#include <utility>
#include <vector>
#include <windows.h>

using namespace std;

struct Movie {
public:
    Movie();
    Movie(string title, int duration, vector<string> genres);

    string title;
    int durationSec;
    vector<string> genres;
    vector<string> reviews;

    void outputTitle() const;

    void outputGenres();

    void outputReviews();

    void output();
};

class Explorer {
public:
    static vector<Movie> movies;

    static void outputMovies();
};

class Finder : public Explorer {
public:
    static Movie findByTitle();
    static void findByGenres();
};

class Player : public Finder {
public:
    static void play();
};

class Upload : public Explorer {
public:
    static void upload();
};

#endif //CODE_FILMLIBRARY_H
```

Содержимое исполняемого файла:

```
#include "FilmLibrary.h"

#include <utility>

void Movie::outputTitle() const {
    cout << "Title: " << title << "\t";
}

void Movie::outputGenres() {
    cout << "Genres: ";
    for (auto &i: genres)
        cout << i << ", ";
}

void Movie::outputReviews() {
    cout << "Reviews: ";
    for (auto &i: reviews)
        cout << i << ", ";
}

void Movie::output() {
    outputTitle();
    outputGenres();
    outputReviews();
    cout << endl;
}

Movie::Movie(string title, int duration, vector<string> genres) {
    this->title = std::move(title);
    this->durationSec = duration;
    this->genres = std::move(genres);
}

Movie::Movie() {
    this->title = "";
    this->durationSec = 0;
    this->genres = {};
}

void Explorer::outputMovies() {
    for (auto &i: movies)
        i.output();
}

Movie Finder::findByTitle() {
    string title;
    cout << "Введите название фильма: ";
    cin >> title;
    for (auto &i: movies)
        if (i.title == title)
            return i;

    Movie error("Not Find", 0, {});
    cout << "Не найдено фильма с введенным названием";
    return error;
}

void Finder::findByGenres() {
    string genre;
```

```

cout << "Введите ваш любимый жанр: ";
cin >> genre;
vector<Movie> res(0);
for (auto &i: movies)
    for (auto &j: i.genres)
        if (genre == j) {
            res.push_back(i);
            break;
        }

if (!res.empty())
    for (auto &i: res)
        i.output();
else
    cout << "Фильмов с вашим любимым жанром не обнаружено :(";
cout << "\n\n";
}

void Player::play() {
    Movie m = findByTitle();
    if (m.title != "Not Find") {
        cout << "Приятного просмотра!\n";
        Sleep(m.durationSec * 1000);
        cout << "Поздравляем!!! Вы посмотрели фильм: " << m.title
            << ". Оставьте отзыв:\n";
        string review;
        cin >> review;
        m.reviews.push_back(review);
    }
}

void Upload::upload() {
    Movie movie;
    cout << "Введите название вашего ролика: ";
    cin >> movie.title;
    cout << "Ну и названице... Хорошо, сколько длиться этот ваш фильм: ";
    cin >> movie.durationSec;
    cout
        << "С длительностью определились. К каким жанрам вы отнесете свою киноленту (Если вы закончили ввод,
напишите \"Все\");";
    while (true) {
        string genre;
        cin >> genre;
        if (genre == "\"Все\"" || genre == "Все") {
            cout << "Ну вот и все. Вы добавили свой... Шедевр! Мы благодарны вам.\n\n";
            break;
        }
        movie.genres.push_back(genre);
    }
    movies.push_back(movie);
}

```

Контрольные вопросы:

1. Что такое модуль?
2. Какие бывают типы модулей в C++?
3. Структура типа «запись» (Struct).
4. Что такое интерфейс?
5. Принципы ООП.