

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: Объектно-ориентированное программирование
тема: «Модульное программирование. Интерфейсы»

Выполнил: ст. группы ПВ-223

Игнатъев Артур Олегович

Проверил:

асс. Черников Сергей Викторович

Белгород 2024г.

Лабораторная работа №2

«Модульное программирование. Интерфейсы»

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

Задание:

1. Разработать программу согласно своего варианта.

Ход выполнения лабораторной работы:

Вариант 3

Разработать программу «Домашняя фильмотека»

Код программы:

Файл interface.h

```
#ifndef OOP_INTERFACE_H
#define OOP_INTERFACE_H

#include <fstream>
#include <string>
#include <iostream>

const int N = 30;

struct info_film {
    char *adress;
    std::string name;
    FILE *f;
    int time;
    int year;

    info_film();

    void set_name(std::string);

    void set_time(int);

    void set_year(int);

    void set_adress(char *);

    void print_name();

    void print_year();

    void print_time();

    void print_all();

    void play_film();

    void off_film();
};
```

```

    void create_film();
};

struct film_collection {
    film_collection();

    void add_film(info_film);

    void dell_film(std::string s);

    int search_film(std::string s);

    void output();

    info_film film[N];
    int n;
};

struct sort_film {

    film_collection film;

    void name_sort();

    void year_sort();
};

struct unity_film {

    void year_unity(info_film t[], int n);
};

struct play {

    film_collection film;

    int run_film(std::string s);

    void run();
};

#endif //OOP_INTERFACE_H

```

Файл interface.cpp

```

#include "interface.h"
#include <iostream>
#include <string>
#include <fstream>
#include <stdlib.h>
#include <locale.h>

using namespace std;

info_film::info_film() {
    name = "";
    time = 0;
    year = 0;
}

```

```

}

void info_film::set_name(std::string s) {
    name = s;
}

void info_film::set_time(int n) {
    time = n;
}

void info_film::set_year(int n) {
    year = n;
}

void info_film::print_name() {
    std::cout <<
    "
    "
    << endl;
    std::cout << "Название фильма: " << name << endl;
}

void info_film::print_time() {
    std::cout << "Продолжительность фильма: " << time << " min" << endl;
}

void info_film::print_year() {
    std::cout << "Год выпуска: " << year << endl;
}

void info_film::print_all() {
    print_name();
    cout << " ";
    print_year();
    cout << " ";
    print_time();
    cout << " ";
}

void info_film::set_adress(char *s) {
    adress = s;
}

void info_film::play_film() {
    cout << "===== ФИЛЬМ =====" << endl;
}

void info_film::off_film() {
    system("CLS");
}

int play::run_film(std::string name) {
    int t = film.search_film(name);
    cout << t;
    if (t != -1)
        film.film[t].play_film();
    else
        cout << "Фильм не найден!" << endl;
    return t;
}

```

```

film_collection::film_collection() {
    n = 3;
    char *s = (char *) "C:\\movies\\1.mp4";
    char *s1 = (char *) "C:\\movies\\2.mp4";
    char *s2 = (char *) "C:\\movies\\3.mp4";

    film[0].set_name("Bleach");
    film[0].set_adress(s);
    film[0].set_year(2004);
    film[0].set_time(8784);

    film[1].set_name("Grand Blue");
    film[1].set_adress(s1);
    film[1].set_year(2018);
    film[1].set_time(288);

    film[2].set_name("Hellsing");
    film[2].set_adress(s2);
    film[2].set_year(2001);
    film[2].set_time(299);
}

void film_collection::add_film(info_film t) {
    if (search_film(t.name) == -1) {
        film[n] = t;
        n++;
    } else
        std::cout << "фильм уже добавлен!" << endl;
}

void film_collection::dell_film(std::string s) {
    int k = search_film(s);
    if (k != -1) {
        film[k] = film[n - 1];
        --n;
    } else
        std::cout << "фильм не найден!" << endl;
}

int film_collection::search_film(std::string s) {
    int i = 0;
    while (i < n) {
        if (film[i].name == s) {
            return i;
        }
        i++;
    }
    return -1;
}

void info_film::create_film() {
    std::string s;

    int n;
    float n1;
    cout << "Введите название : ";
    cin >> s;
    set_name(s);

    cout << "Введите год выпуска: ";
    cin >> n;
}

```

```

        set_year(n);
        cout << "Введите продолжительность: ";
        cin >> n1;
        set_time(n1);
    }

    void sort_film::name_sort() {
        for (int i = 0; i < film.n - 1; i++)
            for (int j = i + 1; j < film.n; j++) {
                if (film.film[i].name > film.film[j].name)
                    std::swap(film.film[i], film.film[j]);
            }
    }

    void sort_film::year_sort() {
        for (int i = 0; i < film.n - 1; i++)
            for (int j = i + 1; j < film.n; j++) {
                if (film.film[i].year > film.film[j].year)
                    std::swap(film.film[i], film.film[j]);
            }
    }

    void unity_film::year_unity(info_film film[], int n) {
        int f = 0, k = 1, i = 0, j = 1;

        std::cout << "----- 1 папка ----- " << endl;
        while (i <= n - 1 && j <= n) {
            if (film[i].year == film[j].year) {
                if (!f) film[i].print_all();
                f = 1;
                film[j].print_all();
                j++;
                continue;
            } else {
                if (f) {
                    k++;
                    std::cout << "----- " << k << " папка -----
" << endl;
                }
                f = 0;
                i = j;
                j = i + 1;
            }
        }

        i = 0;
        j = 1;
        while (i <= n - 1 && j <= n) {
            if (film[i].year == film[j].year) {
                i += 2;
                j += 2;
            } else {
                film[i].print_all();
                i++;
                j++;
            }
        }
    }

    void film_collection::output() {
        for (int i = 0; i < n; i++) {
            film[i].print_all();
        }
    }

```

```

}

void play::run() {
    int n = 0, m, v, f = 0;
    std::string s, s1, y;
    sort_film c;
    while (n != 6) {
        if (!f) c.film.output();
        f = 0;
        cout << "Включить фильм" - 1" << endl;
        cout << "Добавить фильм в фильмотеку" - 2" << endl;
        cout << "Удалить фильм из фильмотеки" - 3" << endl;
        cout << "Отсортировать фильмотеку" - 4" << endl;
        cout << "Объединить фильмы по папкам" - 5" << endl;
        cout << "Выйти из домашней фильмотеки" - 6" << endl;
        cout << "Ваш выбор: ";
        cin >> n;
        switch (n) {
            case 1: {
                cout << "Какой фильм включить? " << endl;
                cin >> s;
                cout << endl;
                int t = run_film(s);
                if (t != -1) {
                    cout << endl;

                    cout << "Выключить фильм? - 1" << endl;

                    c.film.film[t].off_film();
                    break;

                }

                break;
            }
            case 2: {
                info_film t;
                t.create_film();
                c.film.add_film(t);
                break;
            }
            case 3: {
                cout << "Какой фильм удалить? " << endl;
                cin >> s1;
                c.film.dell_film(s1);
                break;
            }
            case 4: {
                cout << "Отсортировать по названию фильма" - 1 " << endl;
                cout << "Отсортировать по году создания" - 2 " << endl;
                cin >> m;
                switch (m) {
                    case 1: {
                        c.name_sort();
                        break;
                    }
                    case 2: {
                        c.year_sort();
                        break;
                    }
                }

                break;
            }
        }
    }
}

```

```

        case 5: {
            cout << "Объединить фильмы в папку по году их создания  - 1"
<< endl;

            cin >> m;

            c.year_sort();
            unity_film u;
            u.year_unity(c.film.film, c.film.n);
            f = 1;

        }
    }
}

```

Файл main.cpp

```

#include <iostream>
#include "../libs/alg/labs/lab2/interface.h"
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include <iostream>
#include<windows.h>
#include<locale.h>

using namespace std;

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(CP_UTF8);
    setlocale(LC_ALL, "Russian");

    play r;
    r.run();
}

```

Результат работы программы:

Название фильма: Bleach

Год выпуска: 2004

Продолжительность фильма: 8784 min

Название фильма: Grand Blue

Год выпуска: 2018

Продолжительность фильма: 288 min

Название фильма: Hellsing

Год выпуска: 2001

Продолжительность фильма: 299 min

Включить фильм - 1

Добавить фильм в фильмотеку - 2

Удалить фильм из фильмотеки - 3

Отсортировать фильмотеку - 4

Объединить фильмы по папкам - 5

Выйти из домашней фильмотеки - 6

Ваш выбор:

Выйти из домашней фильмотеки - 6

Ваш выбор: 2

Введите название : *Naruto*

Введите год выпуска: *2002*

Введите продолжительность: *5060*

Название фильма: Bleach

Год выпуска: 2004

Продолжительность фильма: 8784 min

Название фильма: Grand Blue

Год выпуска: 2018

Продолжительность фильма: 288 min

Название фильма: Hellsing

Год выпуска: 2001

Продолжительность фильма: 299 min

Название фильма: Naruto

Год выпуска: 2002

Продолжительность фильма: 5060 min

Включить фильм

- 1

Вывод: в ходе выполнения лабораторной работы получил навыки модульной декомпозиции предметной области, создания модулей, разработки интерфейсов.