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
тема: « Модульное программирование. Интерфейсы»

Выполнил: ст. группы ПВ202
Аладиб язан
Проверил:
Буханов Д.Г.

Белгород 2022

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++.

Задание Варианта :

Программа «Домашняя библиотека»

Выполнение работы:

interface.h :

```
#ifndef INTERFACE_H
#define INTERFASE_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
```

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(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()
{
    cout <<
"_____
_____" << endl;
    cout << "Movie title: " << name << endl;
}

void info_film::print_time()
{
    cout << "Movie duration: " << time << " min" << endl;
}

void info_film::print_year()
{
    cout << "Release year: " << 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()
// {
//     string command = "start \"\" \"" + string(adress) + "\"";
//     system(command.c_str());
//     cout << "====================== Playing Movie ======================" <<
endl;
// }
void info_film::play_film()
{
    cout << "====================== Playing Movie ======================" <<
endl;
}

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

int play::run_film(string name)
{
    int t = film.search_film(name);
    cout << t;
    if (t != -1)
        film.film[t].play_film();
    else
        cout << "Movie not found!" << endl;
    return t;
}

film_collection::film_collection()
{
    n = 3;
    char* s = (char*)"C:\\Users\\yazon\\Desktop\\oop\\lab2\\fight club.mp4";
    char* s1 = (char*)"C:\\Users\\yazon\\Desktop\\oop\\lab2\\knives out.mp4";
    char* s2 = (char*)"C:\\Users\\yazon\\Desktop\\oop\\lab2\\superbad.mp4";

    film[0].set_name("fight club");
    film[0].set_adress(s);
```

```
    film[0].set_year(1999);
    film[0].set_time(139);

    film[1].set_name("knives out");
    film[1].set_adress(s1);
    film[1].set_year(2019);
    film[1].set_time(130);

    film[2].set_name("superbad");
    film[2].set_adress(s2);
    film[2].set_year(2007);
    film[2].set_time(119);
}

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

void film_collection::dell_film(string s)
{
    int k = search_film(s);
    if (k != -1)
    {
        film[k] = film[n - 1];
        --n;
    }
    else
        cout << "Movie not found!" << endl;
}

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

void info_film::create_film()
```

```
{
    string  s;
    int n;
    float n1;
    cout << "Enter movie title: ";
    cin >> s;
    set_name(s);

    cout << "Enter release year: ";
    cin >> n;
    set_year(n);

    cout << "Enter duration: ";
    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)
                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)
                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;

    cout << "-------------- Folder 1 -------------- " << 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++;
                cout << "-------------- Folder " << k << " -------------- " <<
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;
    string s, s1, y;
    sort_film c;
    while (n != 6)
    {
        if (!f) c.film.output();
        f = 0;
        cout <<
"_____________________________________________________________________________
_________________" << endl;
        cout << "Play movie                                - 1" << endl;
        cout << "Add movie to library                      - 2" << endl;
```

```
cout << "Delete movie from library             - 3" << endl;
cout << "Sort movie library                    - 4" << endl;
cout << "Group movies into folders             - 5" << endl;
cout << "Exit the home movie library           - 6" << endl;
cout << "Your choice: ";
cin >> n;
switch (n)
{
case 1:
{
    cout << "Which movie to play? " << endl;
    cin >> s;
    cout << endl;
    int t = run_film(s);
    if (t != -1) {
        cout << endl;
        cout << "Turn off movie?            - 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 << "Which movie to delete? " << endl;
    cin >> s1;
    c.film.dell_film(s1);
    break;
}
case 4:
{
    cout << "Sort by movie title        - 1 " << endl;
    cout << "Sort by release year       - 2 " << endl;
    cin >> m;
    switch (m)
    {
    case 1:
    {
        c.name_sort();
        break;
    }
    case 2:
    {
        c.year_sort();
        break;
        break;
```

```
            }
            }
            break;
        }
        case 5:
        {
            cout << "Group movies by release year - 1" << endl;
            cin >> m;
            c.year_sort();
            unity_film u;
            u.year_unity(c.film.film, c.film.n);
            f = 1;
        }
        }
    }
}
```

main.cpp :

```
#include <fstream>
#include <iostream>
#include <locale.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
#include "interface.cpp"

using namespace std;

int main(){
    SetConsoleOutputCP(CP_UTF8);//Подключение русского языка
    play r;
    r.run();
}
```