**МИНОБРНАУКИ РОССИИ**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

Дисциплина: Обьектно-ориентированное программирование

по теме «Модульное программирование. Интерфейсы.»

Выполнил: ст. группы ВТ-22  
Ковалев Павел ВТ-22

Проверил: Буханов Д.Г.

Белгород 2019

Вариант 3

**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.

Вариант 3

Программа «Домашняя фильмотека»

<filmlibrary.h>

#ifndef FILMLIBRARY\_H

#define FILMLIBRARY\_H

#include <QTextCodec>

#include <QTextStream>

#include <iostream>

using namespace std;

#include <string>

#include <QObject>

#include <QtDebug>

#include <QString>

#include <stdio.h>

//Основная информация\*\*\*Авторская инфа

struct **general**{

QString doc;

int rate;//1-10;};

//Допольнительная информация\*\*\*С ней можно взаимодействовать

struct **extra**{

int y\_rate;//1-10;};

//Фильм

struct **film**{

QString name;

unsigned pos;

general gen\_info;//Освновная инфа

extra ex\_info;//Доп инфа};

//Фильмы

struct **bib**{

film \*f;//Сами фильмы

unsigned n\_bib;//Количество фильмов};

//Библиотеки

struct **filmbib**{

bib m\_bib;//Подкаталог фильмы

bib fav\_bib;//Избранные фильмы

void **write\_films**();//Выводит всю библиотеку

void **add\_film**();

void **add\_in\_fav**();

void **list\_idFilm**();};

#endif // FILMLIBRARY\_H

<filmlibrary.cpp>

#include "filmlibrary.h"

void filmbib::**list\_idFilm**(){

unsigned i = 0;

while(i<m\_bib.n\_bib){

cout << "Name film:";

qDebug() << m\_bib.f[i].name;

cout << "ID film:" << m\_bib.f[i].pos << endl;

cout << "\n";

i++;

}

}

void filmbib::**add\_in\_fav**(){

unsigned y\_pos;

list\_idFilm();

cout << "Read ID film" << endl;

cin >> y\_pos;

//Временный буффер

filmbib temp;

temp.m\_bib.n\_bib = m\_bib.n\_bib;

temp.fav\_bib.n\_bib = fav\_bib.n\_bib;

temp.m\_bib.f = new film[temp.m\_bib.n\_bib];

temp.fav\_bib.f = new film[temp.fav\_bib.n\_bib];

temp.m\_bib.f = m\_bib.f;

temp.fav\_bib.f = fav\_bib.f;

//Перевыделения памяти

m\_bib.n\_bib--;

fav\_bib.n\_bib++;

m\_bib.f = new film[m\_bib.n\_bib];

fav\_bib.f = new film[fav\_bib.n\_bib];

for (unsigned i = 0; i < temp.fav\_bib.n\_bib; i++) {

fav\_bib.f[i] = temp.fav\_bib.f[i];

}

unsigned i\_id=0;

for (unsigned i = 0;i<temp.m\_bib.n\_bib;i++) {

if(y\_pos!=temp.m\_bib.f[i].pos){

m\_bib.f[i\_id] = temp.m\_bib.f[i];

m\_bib.f[i\_id].pos = i\_id;

i\_id++;

}

}

fav\_bib.f[fav\_bib.n\_bib-1] = temp.m\_bib.f[y\_pos];

}

void filmbib::**write\_films**(){

cout << "Films" << endl;

for (unsigned n=0;n<m\_bib.n\_bib;n++) {

cout << "Name:";

qDebug() << m\_bib.f[n].name;

cout << "Doc:";

qDebug() << m\_bib.f[n].gen\_info.doc;

cout << "Rate:" << m\_bib.f[n].gen\_info.rate << " " << "MRate:" << m\_bib.f[n].ex\_info.y\_rate;

cout << endl;

}

cout << endl;

cout << "Fav-Films" << endl;

for (unsigned n=0;n<fav\_bib.n\_bib;n++) {

cout << "Name:";

qDebug() << fav\_bib.f[n].name;

cout << "Doc:";

qDebug() << fav\_bib.f[n].gen\_info.doc;

cout << "Rate:" << fav\_bib.f[n].gen\_info.rate << " " << "MRate:" << fav\_bib.f[n].ex\_info.y\_rate;

cout << endl;

}

}

void filmbib::**add\_film**(){

//Переменная для сохранение данных

filmbib temp;

temp.m\_bib.n\_bib = m\_bib.n\_bib;

temp.m\_bib.f = new film[temp.m\_bib.n\_bib];

temp.m\_bib.f = m\_bib.f;

//Новая память

m\_bib.n\_bib++;

m\_bib.f = new film[m\_bib.n\_bib];

//Обмен

for (unsigned i = 0; i<temp.m\_bib.n\_bib;i++) {

m\_bib.f[i] = temp.m\_bib.f[i];

}

//Добавление нового фильмы

cout << "Read name new film:";

char str[255];

cin >> str;

m\_bib.f[m\_bib.n\_bib-1].name=str;

cout << "Read doc:";

cin >> str;

m\_bib.f[m\_bib.n\_bib-1].gen\_info.doc = str;

int rate;

cout << "Read rate:";

cin >> rate;

m\_bib.f[m\_bib.n\_bib-1].gen\_info.rate = rate;

cout << "Read mrate:";

cin >> rate;

m\_bib.f[m\_bib.n\_bib-1].ex\_info.y\_rate = rate;

m\_bib.f[m\_bib.n\_bib-1].pos = m\_bib.n\_bib - 1;

}

<main.cpp>

#include <QCoreApplication>

#include "filmlibrary.h"

//Программа «Домашняя фильмотека»

filmbib **save**(){

filmbib save\_film;

save\_film.m\_bib.f = new film[3];

save\_film.m\_bib.n\_bib = 3;

save\_film.fav\_bib.f = new film[1];

save\_film.fav\_bib.n\_bib = 1;

save\_film.m\_bib.f[0].name = "The Shawshank Redemption";

save\_film.m\_bib.f[0].gen\_info.doc = '1';

save\_film.m\_bib.f[0].pos = 0;

save\_film.m\_bib.f[0].gen\_info.rate = 3;

save\_film.m\_bib.f[0].ex\_info.y\_rate = 2;

save\_film.m\_bib.f[1].name = "The Green Mile";

save\_film.m\_bib.f[1].gen\_info.doc = "2";

save\_film.m\_bib.f[1].pos = 1;

save\_film.m\_bib.f[1].gen\_info.rate = 5;

save\_film.m\_bib.f[1].ex\_info.y\_rate = 6;

save\_film.m\_bib.f[2].name = "Forrest Gump";

save\_film.m\_bib.f[2].gen\_info.doc = "3";

save\_film.m\_bib.f[2].pos = 2;

save\_film.m\_bib.f[2].gen\_info.rate = 1;

save\_film.m\_bib.f[2].ex\_info.y\_rate = 2;

save\_film.fav\_bib.f[0].name = "Schindler's List";

save\_film.fav\_bib.f[0].gen\_info.doc = "4";

save\_film.fav\_bib.f[0].pos = 0;

save\_film.fav\_bib.f[0].gen\_info.rate = 3;

save\_film.fav\_bib.f[0].ex\_info.y\_rate = 9;

return save\_film;

}

int **main**(int argc, char \*argv[])

{

QCoreApplication applications(*argc*, argv);

#ifdef Q\_OS\_WIN32

QTextCodec::setCodecForLocale(QTextCodec::codecForName("IBM 866"));

#endif

#ifdef Q\_OS\_LINUX

QTextCodec::setCodecForLocale(QTextCodec::codecForName("UTF-8"));

#endif

filmbib savefilm = save();

savefilm.add\_in\_fav();

savefilm.write\_films();

return applications.exec();

}