МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

И ПРОГРАММНОЙ ИНЖЕНЕРИИ (КАФЕДРА №43)

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Е. В. Павлов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

«ПРЯМОЕ И ОБРАТНОЕ ПРОЕКТИРОВАНИЕ»

ПО ДИСЦИПЛИНЕ «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ

ИНФОРМАЦИОННЫХ СИСТЕМ»

­РАБОТУ ВЫПОЛНИЛ (ЛА)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ (КА) ГР. | 4631 |  |  |  | С.А. Гришин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург   
2018

1. **Цель работы**

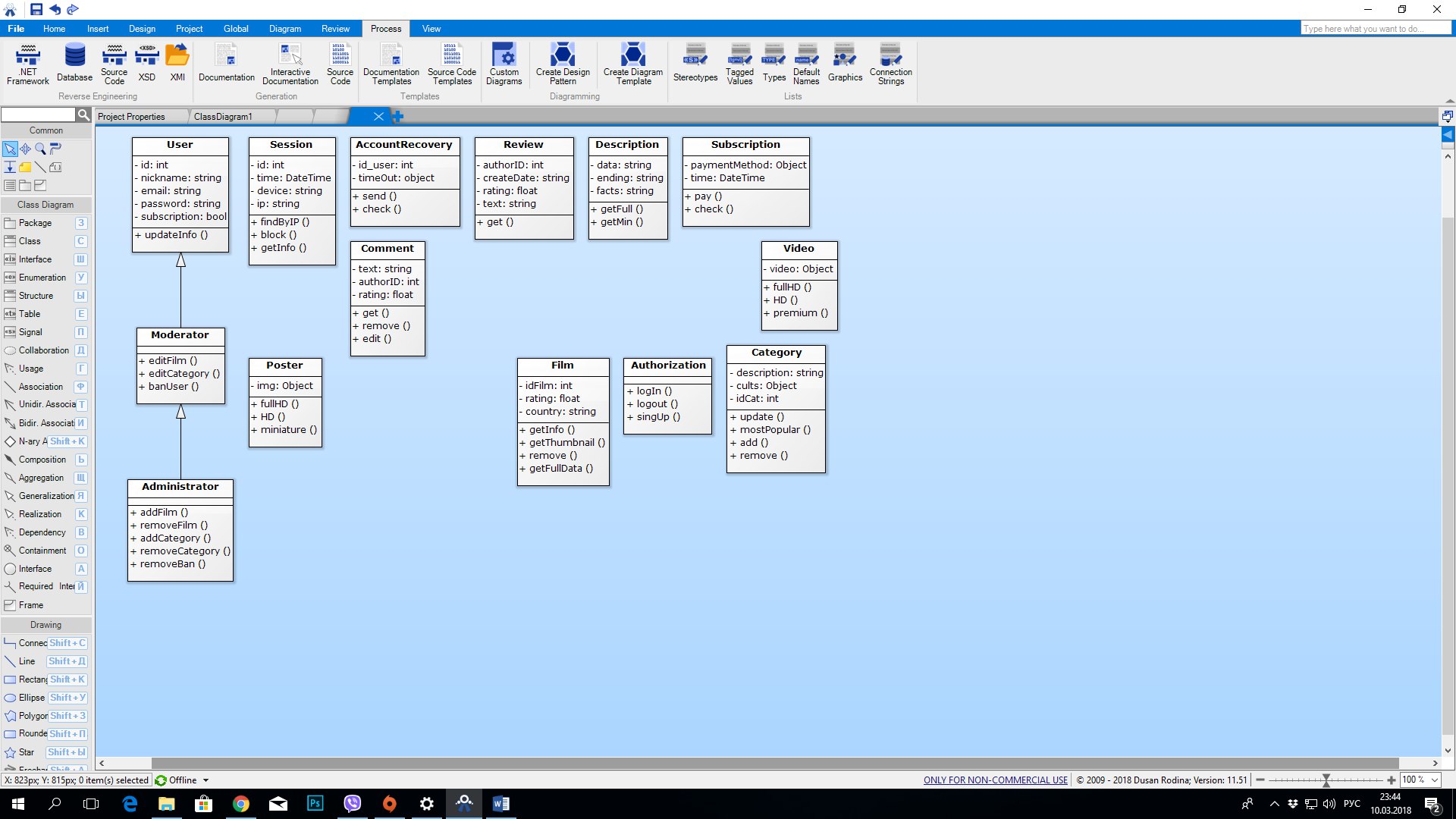
Целью данной работы является получение навыков генерации кода для UML-модели и анализ программного обеспечения через обратное проектирование.

1. **Задание**

По разработанной в ЛР №2 диаграмме классов осуществить генерацию кода в любой объектно-ориентированный язык и выполнить обратное проектирование полученного кода. Сравнить исходную и полученную модель.

**Вариант 75: База данных о фильмах**

1. **Диаграмма вариантов использования**

****

1. **Выводы**

В результате выполнения данной лабораторной работы были получены практические навыки обратного проектирования и генерации кода из UML-модели.

Исходная диаграмма классов была разработана в ЛР №2 для системы «База данных о фильмах».

Полученная в результате обратного проектирования диаграмма классов, отличается от исходной, в частности пропали все ассоциации, реализации, зависимости и агрегации, осталось только наследование. Данное отличие связано с тем,

что некоторая информация на исходной диаграмме классов находится на более высоком

уровне детализации, который выходит за рамки семантики языка реализации.

Соответственно при генерации кода данная информация была утеряна, за исключением отношений наследования.

Генерация кода и обратное проектирование проводились в Software Ideas Modeler при стандартных настройках (по умолчанию).

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Software Ideas Modeler [Электронный ресурс].– URL: <https://www.softwareideas.net/>
2. Буч Грэди Введение в UML от создателей языка / Грэди Буч, Джеймс Рамбо, Айвар Якобсон: пер. с англ. – ДМК Пресс, 2015 – 496 с.: ил.

1. Ларман Крэг Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку: пер. с англ. – М.: ИД "Вильямс", 2013. – 736 с.: ил.

**Приложение**

class AccountRecovery

{

private:

int id\_user;

object timeOut;

public:

void send ();

void check ();

};

void AccountRecovery::send ()

{

}

void AccountRecovery::check ()

{

}

class Session

{

private:

int id;

DateTime time;

string device;

string ip;

public:

void findByIP ();

void block ();

void getInfo ();

};

void Session::findByIP ()

{

}

void Session::block ()

{

}

void Session::getInfo ()

{

}

class Review

{

private:

int authorID;

string createDate;

float rating;

string text;

public:

void get ();

};

void Review::get ()

{

}

class Comment

{

private:

string text;

int authorID;

float rating;

public:

void get ();

void remove ();

void edit ();

};

void Comment::get ()

{

}

void Comment::remove ()

{

}

void Comment::edit ()

{

}

class Poster

{

private:

Object img;

public:

void fullHD ();

void HD ();

void miniature ();

};

void Poster::fullHD ()

{

}

void Poster::HD ()

{

}

void Poster::miniature ()

{

}

class Film

{

private:

int idFilm;

float rating;

string country;

public:

void getInfo ();

void getThumbnail ();

void remove ();

void getFullData ();

};

void Film::getInfo ()

{

}

void Film::getThumbnail ()

{

}

void Film::remove ()

{

}

void Film::getFullData ()

{

}

class Description

{

private:

string data;

string ending;

string facts;

public:

void getFull ();

void getMin ();

};

void Description::getFull ()

{

}

void Description::getMin ()

{

}

class Authorization

{

public:

void logIn ();

void logout ();

void singUp ();

};

void Authorization::logIn ()

{

}

void Authorization::logout ()

{

}

void Authorization::singUp ()

{

}

class Subscription

{

private:

Object paymentMethod;

DateTime time;

public:

void pay ();

void check ();

};

void Subscription::pay ()

{

}

void Subscription::check ()

{

}

class Video

{

private:

Object video;

public:

void fullHD ();

void HD ();

void premium ();

};

void Video::fullHD ()

{

}

void Video::HD ()

{

}

void Video::premium ()

{

}

class Moderator : public User

{

public:

void editFilm ();

void editСategory ();

void banUser ();

};

void Moderator::editFilm ()

{

}

void Moderator::editСategory ()

{

}

void Moderator::banUser ()

{

}

class Administrator : public Moderator

{

public:

void addFilm ();

void removeFilm ();

void addСategory ();

void removeСategory ();

void removeBan ();

};

void Administrator::addFilm ()

{

}

void Administrator::removeFilm ()

{

}

void Administrator::addСategory ()

{

}

void Administrator::removeСategory ()

{

}

void Administrator::removeBan ()

{

}

class User

{

private:

int id;

string nickname;

string email;

string password;

bool subscription;

public:

void updateInfo ();

};

void User::updateInfo ()

{

}

class Category

{

private:

string description;

Object cults;

int idCat;

public:

void update ();

void mostPopular ();

void add ();

void remove ();

};

void Category::update ()

{

}

void Category::mostPopular ()

{

}

void Category::add ()

{

}

void Category::remove ()

{

}

class PremiumAccess :

{

public:

void getAnyQuality ();

void writeComment ();

};

void PremiumAccess::getAnyQuality ()

{

}

void PremiumAccess::writeComment ()

{

}

class DemoInterface :

{

public:

void getHD ();

};

void DemoInterface::getHD ()

{

}