

Федеральное государственное автономное образовательное учреждение  
высшего образования «**Национальный исследовательский университет**  
**ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники  
Дисциплина: «Информационные Системы»

**Отчет**  
По лабораторной работе №1  
Вариант 12222

**Преподаватель:**  
Тюрин Иван Николаевич

**Выполнила:**  
Черемисова Мария Александровна

**Группа:** Р3310

Санкт-Петербург 2025

# Содержание

1. Задание .....	2
2. UML-диаграммы классов и пакетов разработанного приложения .....	6
3. Ссылка на репозиторий с исходным кодом .....	8
4. Вывод .....	8

## 1. Задание

Реализовать информационную систему, которая позволяет взаимодействовать с объектами класса Movie, описание которого приведено ниже:

```
public class Movie {  
    private long id; //Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно генерироваться автоматически  
    private String name; //Поле не может быть null, Стока не может быть пустой  
    private Coordinates coordinates; //Поле не может быть null  
    private java.time.LocalDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически  
    private int oscarsCount; //Значение поля должно быть больше 0  
    private Integer budget; //Значение поля должно быть больше 0, Поле не может быть null  
    private long totalBoxOffice; //Значение поля должно быть больше 0  
    private MpaaRating mpaaRating; //Поле не может быть null  
    private Person director; //Поле не может быть null  
    private Person screenwriter;  
    private Person operator; //Поле не может быть null  
    private Integer length; //Поле может быть null, Значение поля должно быть больше 0  
    private long goldenPalmCount; //Значение поля должно быть больше 0  
    private long usaBoxOffice; //Значение поля должно быть больше 0  
    private MovieGenre genre; //Поле не может быть null  
}  
public class Coordinates {  
    private double x; //Значение поля должно быть больше -553  
    private Double y; //Поле не может быть null  
}  
public class Person {  
    private String name; //Поле не может быть null, Стока не может быть пустой
```

```

    private Color eyeColor; //Поле не может быть null
    private Color hairColor; //Поле может быть null
    private Location location; //Поле может быть null
    private String passportID; //Значение этого поля должно быть
уникальным, Длина строки не должна быть больше 26, Длина строки
должна быть не меньше 10, Поле может быть null
    private Country nationality; //Поле может быть null
}

public class Location {
    private float x; //Поле не может быть null
    private int y;
    private String name; //Поле может быть null
}

public enum MpaaRating {
    G,
    PG,
    PG_13,
    R
}

public enum MovieGenre {
    WESTERN,
    ADVENTURE,
    TRAGEDY,
    HORROR,
    SCIENCE_FICTION
}

public enum Color {
    RED,
    BLACK,
    BLUE,
    ORANGE
}

public enum Country {
    USA,
    GERMANY,
    SPAIN,
    CHINA,
    VATICAN
}

```

Разработанная система должна удовлетворять следующим требованиям:

- Основное назначение информационной системы - управление объектами, созданными на основе заданного в варианте класса.

- Необходимо, чтобы с помощью системы можно было выполнить следующие операции с объектами: создание нового объекта, получение информации об объекте по ИД, обновление объекта (модификация его атрибутов), удаление объекта. Операции должны осуществляться в отдельных окнах (интерфейсах) приложения. При получении информации об объекте класса должна также выводиться информация о связанных с ним объектах.
- При создании объекта класса необходимо дать пользователю возможность связать новый объект с объектами вспомогательных классов, которые могут быть связаны с созданным объектом и уже есть в системе.
- Выполнение операций по управлению объектами должно осуществляться на серверной части (не на клиенте), изменения должны синхронизироваться с базой данных.
- На главном экране системы должен выводиться список текущих объектов в виде таблицы (каждый атрибут объекта - отдельная колонка в таблице). При отображении таблицы должна использоваться пагинация (если все объекты не помещаются на одном экране).
- Нужно обеспечить возможность фильтровать/сортировать строки таблицы, которые показывают объекты (по значениям любой из строковых колонок). Фильтрация элементов должна производиться по неполному совпадению.
- Переход к обновлению (модификации) объекта должен быть возможен из таблицы с общим списком объектов и из области с визуализацией объекта (при ее реализации).
- При добавлении/удалении/изменении объекта, он должен автоматически появиться/исчезнуть/измениться в интерфейсах у других пользователей, авторизованных в системе.
- Если при удалении объекта с ним связан другой объект, операция должна быть отменена, пользователю нужно сообщить о невозможности удаления объекта.
- Пользователи должны иметь возможность просмотра всех объектов. Для модификации объекта должно открываться отдельное диалоговое окно. При вводе некорректных значений в поля объекта должны появляться информативные сообщения о соответствующих ошибках.

В системе должен быть реализован отдельный пользовательский интерфейс для выполнения специальных операций над объектами:

- Вернуть количество объектов, значение поля goldenPalmCount которых равно заданному.
- Вернуть количество объектов, значение поля usaBoxOffice которых больше заданного.
- Вернуть массив уникальных значений поля genre по всем объектам.
- Добавить всем фильмам категории «R» по одному «Оскару».
- Отобрать все «Оскары» у всех фильмов режиссёров, снявших хоть один фильм в указанном жанре.
- Представленные операции должны быть реализованы в качестве функций БД, которые необходимо вызывать из уровня бизнес-логики приложения.

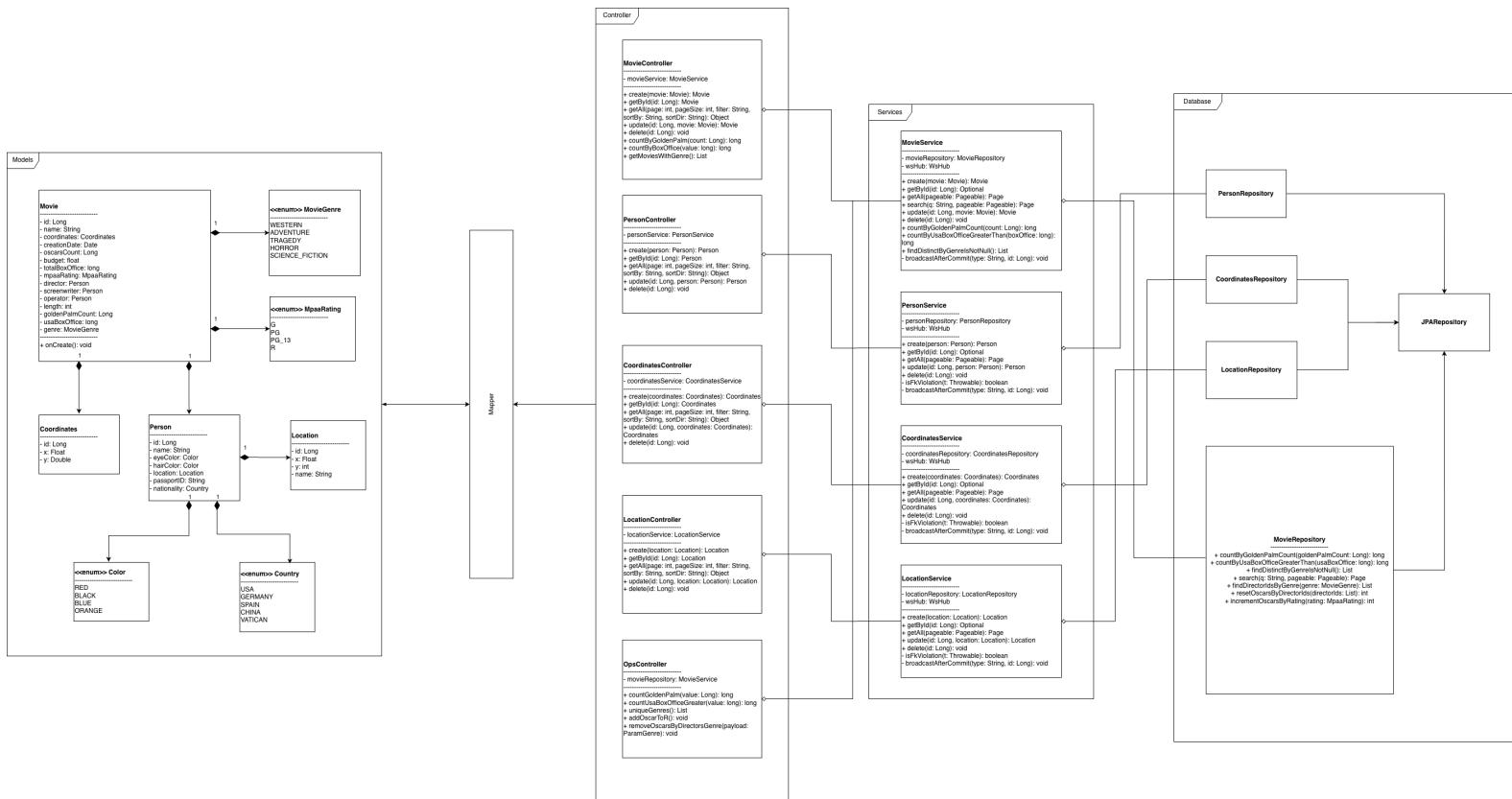
Особенности хранения объектов, которые должны быть реализованы в системе:

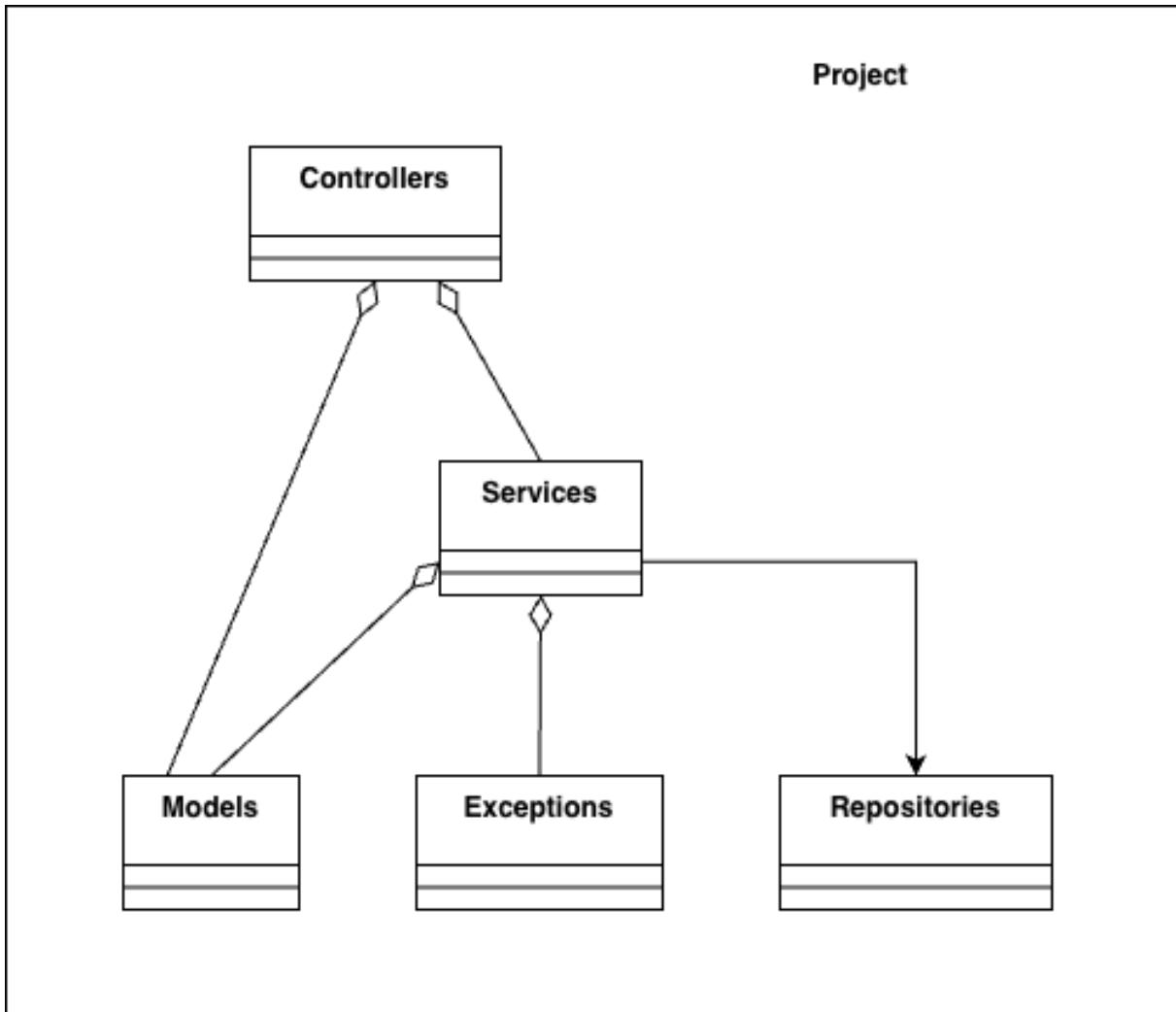
- Организовать хранение данных об объектах в реляционной СУБД (PostgreSQL). Каждый объект, с которым работает ИС, должен быть сохранен в базе данных.
- Все требования к полям класса (указанные в виде комментариев к описанию классов) должны быть выполнены на уровне ORM и БД.
- Для генерации поля id использовать средства базы данных.
- Для подключения к БД на кафедральном сервере использовать хост pg, имя базы данных - studs, имя пользователя/пароль совпадают с таковыми для подключения к серверу.
- При создании системы нужно учитывать следующие особенности организации взаимодействия с пользователем:
- Система должна реагировать на некорректный пользовательский ввод, ограничивая ввод недопустимых значений и информируя пользователей о причине ошибки.
- Переходы между различными логически обособленными частями системы должны осуществляться с помощью меню.
- При добавлении/удалении/изменении объекта, он должен автоматически появиться/исчезнуть/измениться на области у всех других клиентов.

При разработке ИС должны учитываться следующие требования:

- В качестве основы для реализации ИС необходимо использовать Spring MVC.
- Для создания уровня хранения необходимо использовать JPA + EclipseLink.
- Разные уровни приложения должны быть отделены друг от друга, разные логические части ИС должны находиться в отдельных

## 2. UML-диаграммы классов и пакетов разработанного приложения





### **3. Ссылка на репозиторий с исходным кодом**

GitHub: <https://github.com/hipeoplea/infom-systems-lab1/tree/dev>

### **4. Вывод**

В ходе лабораторной работы разобралась в работе таких технологий , как Spring MVC, Spring Data. Вспомнила основные методы запросов, реализовала хранение объектов в БД при помощи ORM провайдера Eclipse. Реализовала работу с объектами при помощи http запросов, сделала выполнение операций асинхронными.