Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет ИТМО»

*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.01 - Информатика и вычислительная техника, Компьютерные системы и технологии*

*Дисциплина «Программирование»*

**Отчет**

**По лабораторной работе №7**

**Вариант №3131703**

Выполнил:

Колмаков Дмитрий Владимирович,

группа Р3131

Преподаватель:

Письмак Алексей Евгеньевич

Г. Санкт-Петербург, 2023 г.

Оглавление

[Текст задания 3](#_Toc135233156)

[Даталогическая модель 4](#_Toc135233157)

[UML-диаграмма классов 5](#_Toc135233158)

[Код программы 6](#_Toc135233159)

[Заключение 7](#_Toc135233160)

# Текст задания

Доработать программу из [лабораторной работы №6](https://se.ifmo.ru/courses/programming#lab6) следующим образом:

1. Организовать хранение коллекции в реляционной СУБД (PostgresQL). Убрать хранение коллекции в файле.
2. Для генерации поля id использовать средства базы данных (sequence).
3. Обновлять состояние коллекции в памяти только при успешном добавлении объекта в БД
4. Все команды получения данных должны работать с коллекцией в памяти, а не в БД
5. Организовать возможность регистрации и авторизации пользователей. У пользователя есть возможность указать пароль.
6. Пароли при хранении хэшировать алгоритмом MD2
7. Запретить выполнение команд не авторизованным пользователям.
8. При хранении объектов сохранять информацию о пользователе, который создал этот объект.
9. Пользователи должны иметь возможность просмотра всех объектов коллекции, но модифицировать могут только принадлежащие им.
10. Для идентификации пользователя отправлять логин и пароль с каждым запросом.

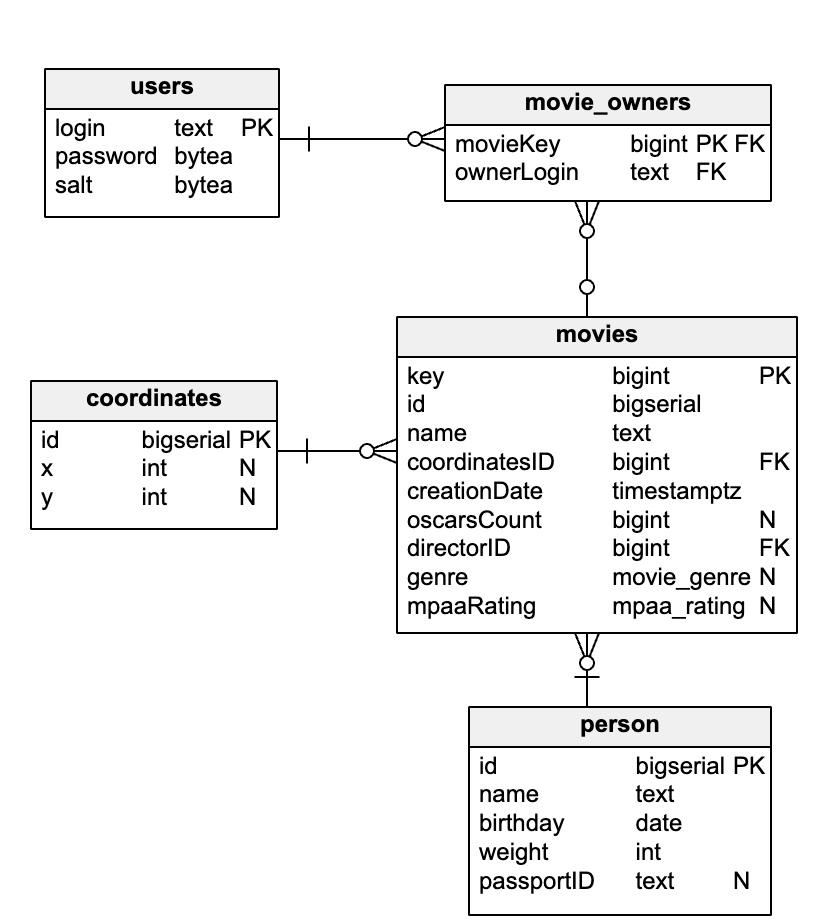
Необходимо реализовать многопоточную обработку запросов.

1. Для многопоточного чтения запросов использовать создание нового потока (java.lang.Thread)
2. Для многопотчной обработки полученного запроса использовать создание нового потока (java.lang.Thread)
3. Для многопоточной отправки ответа использовать ForkJoinPool
4. Для синхронизации доступа к коллекции использовать синхронизацию чтения и записи с помощью synchronized

**Порядок выполнения работы:**

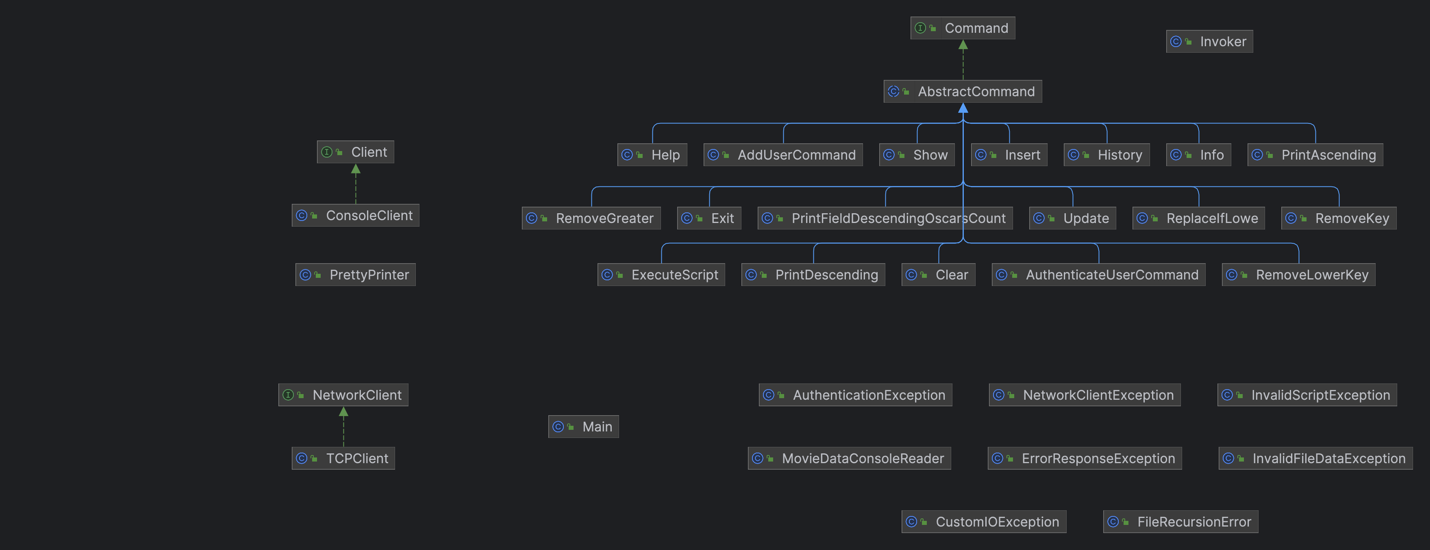
1. В качестве базы данных использовать PostgreSQL.
2. Для подключения к БД на кафедральном сервере использовать хост pg, имя базы данных - studs, имя пользователя/пароль совпадают с таковыми для подключения к серверу.

# Даталогическая модель

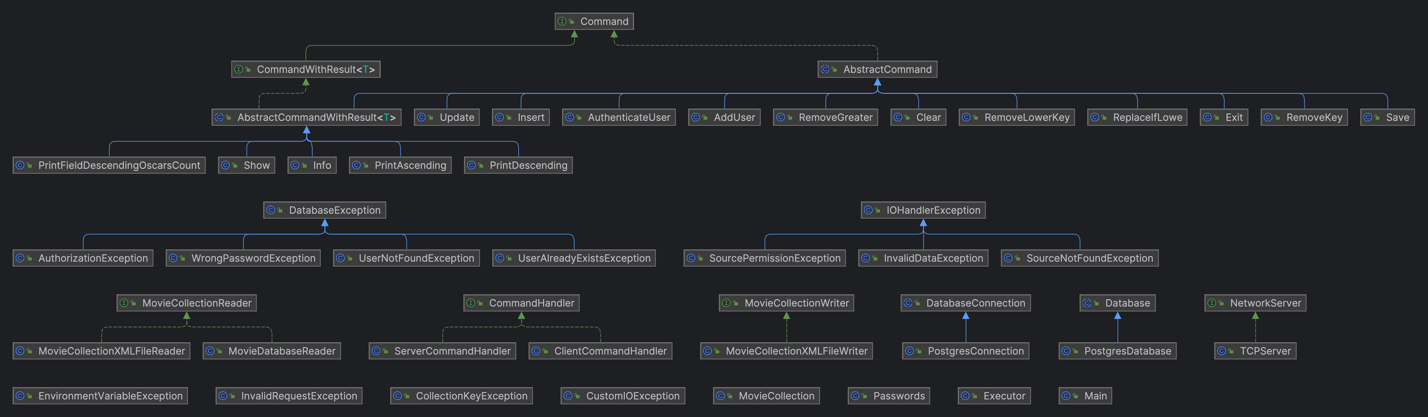


# UML-диаграмма классов

Client



Server



Common



# Код программы

Код программы доступен по ссылке: [https://github.com/whatever125/programming/tree/main/lab7/](https://github.com/whatever125/programming/tree/main/lab7).

# Заключение

В ходе выполнения данной лабораторной работы я успешно применил свои знания, полученные из курса Баз данных. Спроектировал базу данных на PostgreSQL, разработав даталогическую модель.

В процессе работы, я изучил основы взаимодействия с реляционными базами данных на языке Java, используя JDBC API. Я разработал и выполнил SQL-запросы для манипуляции данными в базе данных и обработал результаты с использованием объектов ResultSet.

Также реализовал многопоточную работу программы используя класс Thread и ForkJoinPool. Это позволило мне оптимизировать выполнение задач и улучшить производительность приложения.

В результате этой лабораторной работы, я получил практические навыки работы с базами данных, изучил использование JDBC API и научился применять многопоточность для оптимизации работы программы. Эти знания и опыт помогут мне в дальнейшем развитии как программиста и специалиста в области баз данных.