МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №7

по дисциплине

«ПРОГРАММИРОВАНИЕ»

Вариант № 31160901

Выполнил:

Шпинева Ульяна Сергеевна, P3116

Преподаватель:

Письмак Алексей Евгеньевич

Санкт-Петербург, 2023

Оглавление

[Задание 3](#_Toc135862871)

[Диаграмма классов реализованной объектной модели 4](#_Toc135862872)

[Исходный код программы 4](#_Toc135862873)

[Вывод 5](#_Toc135862874)

# Задание

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

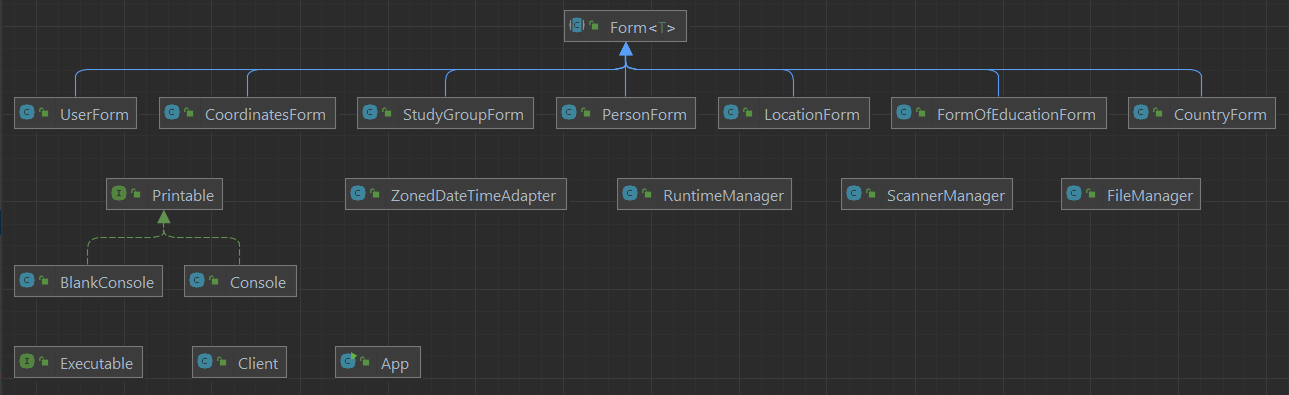
1. Организовать хранение коллекции в реляционной СУБД (PostgresQL). Убрать хранение коллекции в файле.
2. Для генерации поля id использовать средства базы данных (sequence).
3. Обновлять состояние коллекции в памяти только при успешном добавлении объекта в БД
4. Все команды получения данных должны работать 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 SHA-1
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 ForkJoinPool
2. Для многопотчной обработки полученного запроса использовать Fixed thread pool
3. Для многопоточной отправки ответа использовать создание нового потока (java.lang.Thread)
4. Для синхронизации доступа к коллекции использовать синхронизацию чтения и записи с помощью synchronized

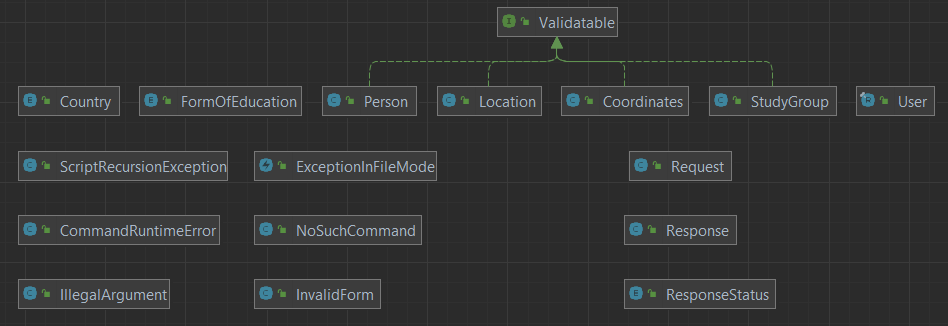
# Диаграмма классов реализованной объектной модели

Client



Server

Common



# Исходный код программы

<https://github.com/UlianaShpineva/Prog_lab7/tree/master>

# Вывод

В ходе лабораторной работы я ознакомилась с основами многопоточного программирования в Java, научилась работать с базами данных PostgeSQL с использованием JDBC.