Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

Высшего образования

*Факультет Программной Инженерии и Компьютерной Техники*

**Лабораторная работа 7 по программированию**

Многопоточность и базы данных

Вариант №31160910

Группа: P3116

Выполнил:

Сиразетдинов А.Н.

Проверил:

Письмак А.Е.

Г. Санкт-Петербург

2023

Оглавление

[Текст задания 3](#_Toc133446150)

[Диаграмма классов разработанной программы 4](#_Toc133446151)

[Клиент 4](#_Toc133446152)

[Общее 4](#_Toc133446153)

[Сервер 5](#_Toc133446154)

[Исходный код программы. 6](#_Toc133446155)

[Вывод 7](#_Toc133446156)

# Текст задания

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

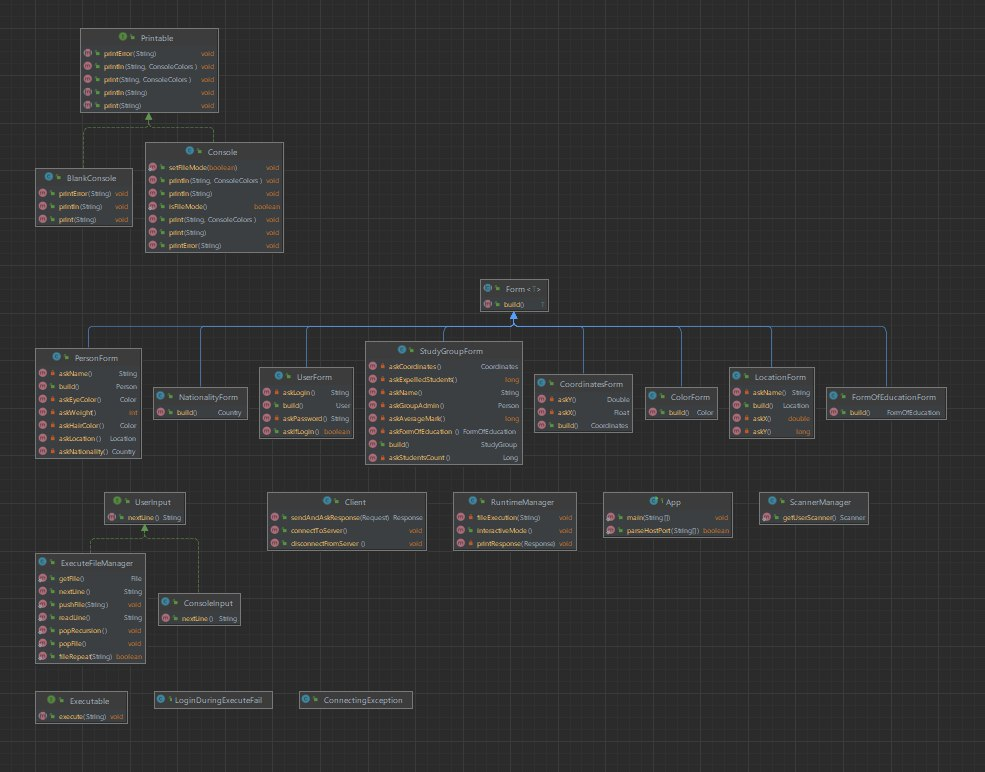
1. Организовать хранение коллекции в реляционной СУБД (PostgresQL). Убрать хранение коллекции в файле.
2. Для генерации поля id использовать средства базы данных (sequence).
3. Обновлять состояние коллекции в памяти только при успешном добавлении объекта в БД
4. Все команды получения данных должны работать 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 SHA-384
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.

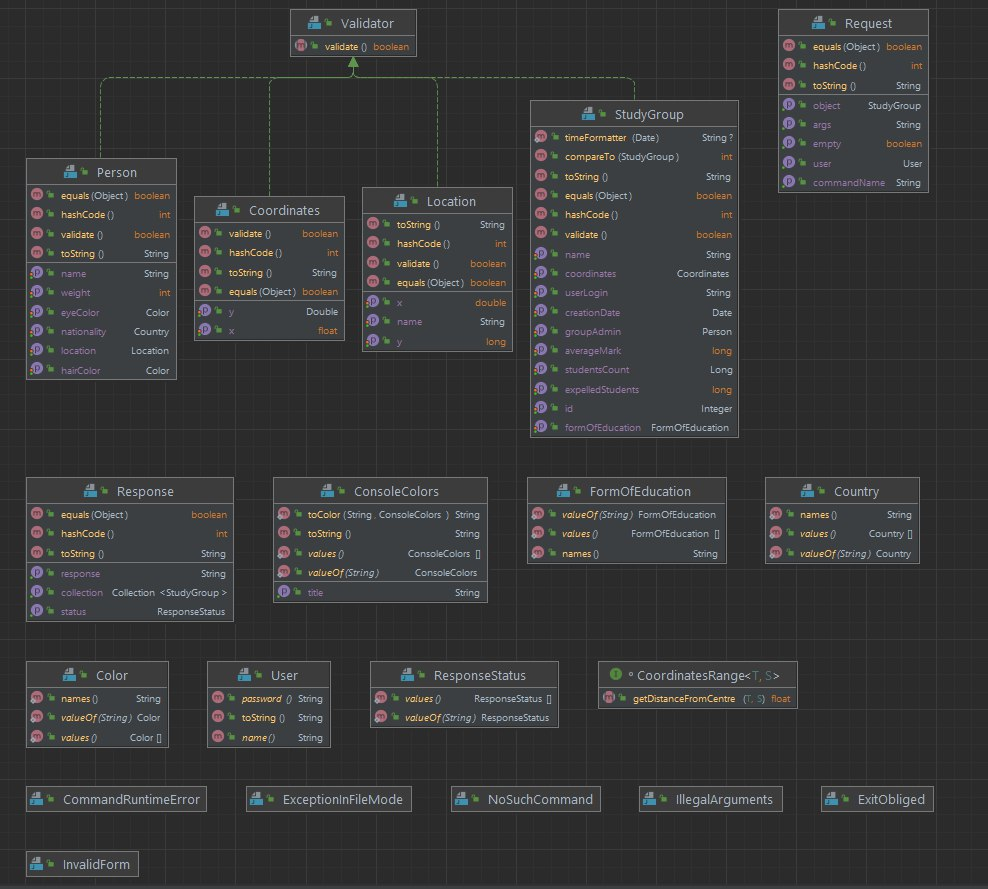
1. Для многопоточного чтения запросов использовать Cached thread pool
2. Для многопотчной обработки полученного запроса использовать Fixed thread pool
3. Для многопоточной отправки ответа использовать ForkJoinPool
4. Для синхронизации доступа к коллекции использовать синхронизацию чтения и записи с помощью java.util.concurrent.locks.ReadWriteLock

# Диаграмма классов разработанной программы

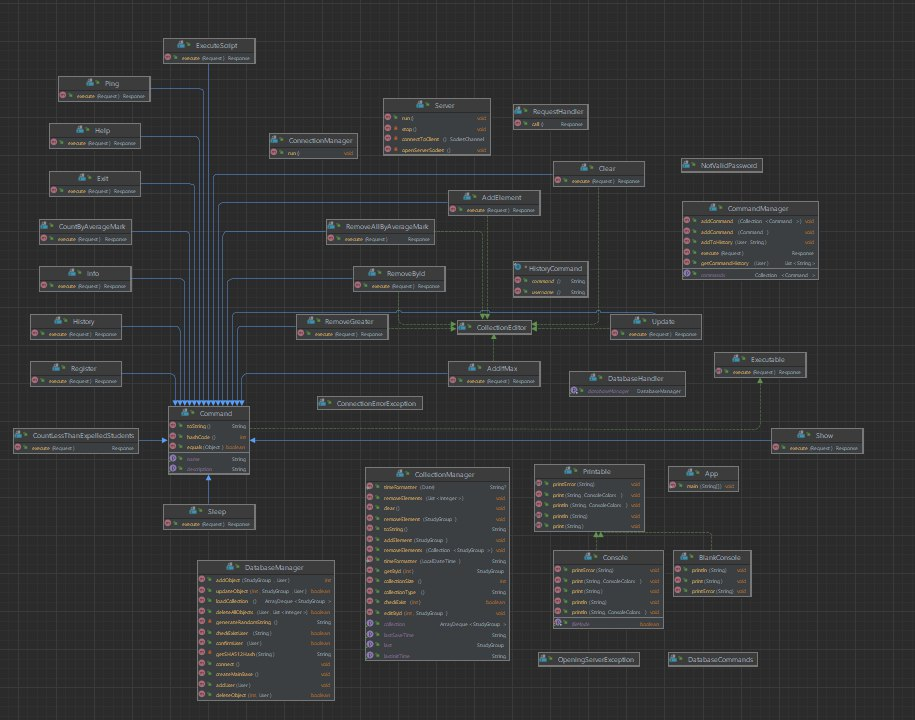
## Клиент



## Общее



## Сервер



# Исходный код программы.

<https://github.com/Azat2202/Prog_lab7>

# Вывод

В процессе выполнения лабораторной работы я расширил возможности своего консольного приложения. Я научился работать с многопоточностью и базами данных.