

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»
Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа №7

по дисциплине

«Программирование»

Вариант №367581.97

Выполнил студент группы Р3115
Федоров Егор Владимирович
Преподаватель:
Кустарев Иван Павлович

1 Текст задания

Доработать программу из лабораторной работы №6 следующим образом:

- Организовать хранение коллекции в реляционной СУБД (PostgresQL). Убрать хранение коллекции в файле.
- Для генерации поля id использовать средства базы данных (sequence).
- Обновлять состояние коллекции в памяти только при успешном добавлении объекта в БД
- Все команды получения данных должны работать с коллекцией в памяти, а не в БД
- Организовать возможность регистрации и авторизации пользователей. У пользователя есть возможность указать пароль.
- Пароли при хранении хэшировать алгоритмом SHA-256
- Запретить выполнение команд не авторизованным пользователям.
- При хранении объектов сохранять информацию о пользователе, который создал этот объект.
- Пользователи должны иметь возможность просмотра всех объектов коллекции, но модифицировать могут только принадлежащие им.
- Для идентификации пользователя отправлять логин и пароль с каждым запросом.

Необходимо реализовать многопоточную обработку запросов.

- Для многопоточного чтения запросов использовать Fixed thread pool
- Для многопоточной обработки полученного запроса использовать создание нового потока (java.lang.Thread)
- Для многопоточной отправки ответа использовать создание нового потока (java.lang.Thread)
- Для синхронизации доступа к коллекции использовать синхронизацию чтения и записи с помощью java.util.concurrent.locks.ReentrantLock

2 Исходный код программы

Исходный код доступен в git-репозитории по адресу <https://github.com/FEgor04/labs/tree/main/programming/lab7>

3 Вывод

Во время выполнения данной лабораторной работы я научился работать с классами `java.io.InputStreamReader`, `java.io.OutputStreamWriter`, закрепил знания ООП и SOLID на практике, изучил возможности `gradle` по тестированию с помощью библиотек `JUnit5` и `mockk`, реализовал генерацию отчета о покрытии с помощью `jacoco` и генерацию HTML-документации с помощью плагина `dokka`.