

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

Факультет программной инженерии и компьютерной техники

Отчет

По лабораторной работе №7

По дисциплине «Программирование»

Вариант 4112

Выполнил: Чураков Александр Алексеевич,
группа Р3131

Преподаватель: Абузов Ярослав Александрович

Санкт-Петербург

~ 2024 ~

Задание

Доработать программу из [лабораторной работы №6](#) 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 `создание нового потока (java.lang.Thread)`
2. Для многопоточной обработки полученного запроса использовать `cached 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`

Исходный код

[ITMO_Labs/ProgrammingJava, Программирование/lab7 at main · awesoma31/ITMO_Labs \(github.com\)](https://github.com/awesoma31/ITMO_Labs/ProgrammingJava)

Вывод

Во время выполнения данной лабораторной работы я научился разрабатывать многопоточные сервера, способные обрабатывать множество клиентов параллельно.