

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «Санкт-Петербургский национальный исследовательский
университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ
ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №7

по дисциплине

«ПРОГРАММИРОВАНИЕ»

Вариант № 31160913

Выполнил:

Студент группы Р3116

Билошицкий Михаил Владимирович

Преподаватель:

Гаврилов Антон Валерьевич

Санкт-Петербург, 2023

Содержание

Задание.....	3
Исходный код программы	4
UML диаграмма классов	5
Вывод.....	6

Задание

Лабораторная работа #7

Введите вариант: 3116C

Внимание! У разных вариантов разный текст задания!

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 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 ForkJoinPool
2. Для многопоточной 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 java.util.concurrent.locks.ReentrantLock

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.

Отчёт по работе должен содержать:

1. Текст задания.
2. Диаграмма классов разработанной программы.
3. Исходный код программы.
4. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Многопоточность. Класс Thread, интерфейс Runnable. Модификатор synchronized.
2. Методы wait(), notify() класса Object, интерфейсы Lock и Condition.
3. Классы-синхронизаторы из пакета java.util.concurrent.
4. Модификатор volatile. Атомарные типы данных и операции.
5. Коллекции из пакета java.util.concurrent.
6. Интерфейсы Executor, ExecutorService, Callable, Future
7. Пулы потоков
8. JDBC. Порядок взаимодействия с базой данных. Класс DriverManager. Интерфейс Connection
9. Интерфейсы Statement, PreparedStatement, ResultSet, RowSet
10. Шаблоны проектирования.

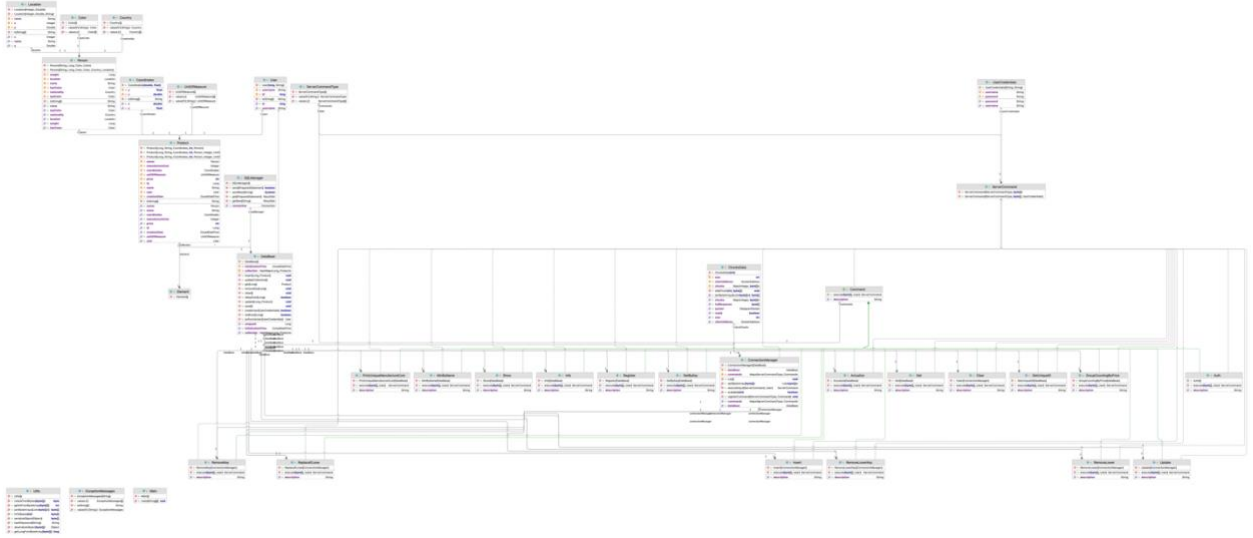
Исходный код программы

Ссылка на GitHub с исходным кодом:

https://github.com/michael-bill/labs_programming_itmo/tree/main/laba7

UML диаграмма классов

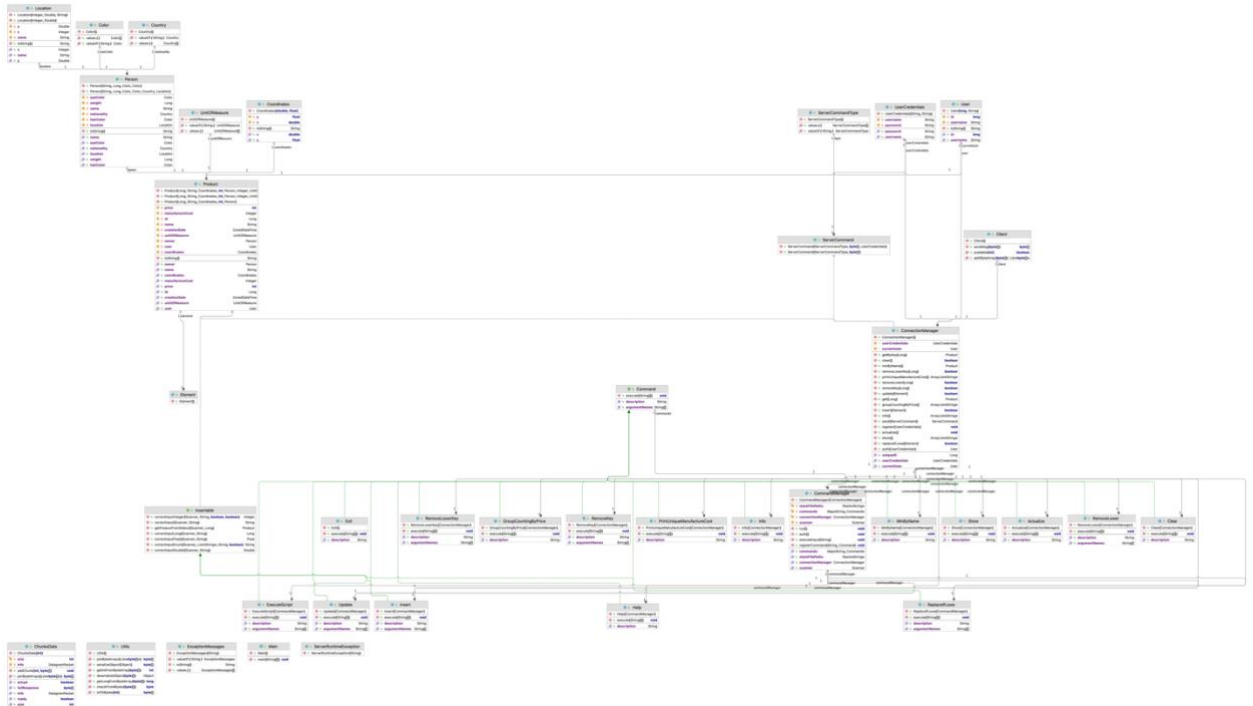
Серверное приложение.



Чтобы ознакомиться с ней поближе, ссылка на .png файл также прикреплена в GitHub репозитории:

https://github.com/michael-bill/labs_programming_itmo/blob/main/laba7/laba7_server_diagram.png

Клиентское приложение.



Чтобы ознакомиться с ней поближе, ссылка на .png файл также прикреплена в GitHub репозитории.

https://github.com/michael-bill/labs_programming_itmo/blob/main/laba7/laba7_client_diagram.png

Вывод

В ходе выполнения лабораторной работы я научился писать клиент серверное приложение с многопоточной обработкой данных на сервере, с грамотным распределением потоков и сохранением целостности данных коллекции, научился реализовывать авторизацию в приложении с шифрованием пароля различными алгоритмами, а также хранением данных о пользователях и элементах коллекции в базе данных PostgreSQL. Полученные знания пригодятся мне в будущем.