Факультет программной инженерии и компьютерной техники

Программирование

Лабораторная работа №7

Вариант 311301

Преподаватель: Николаев Владимир Вячеславович

Выполнил: Кульбако Артемий Юрьевич

Р3112

# Текст задания.

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

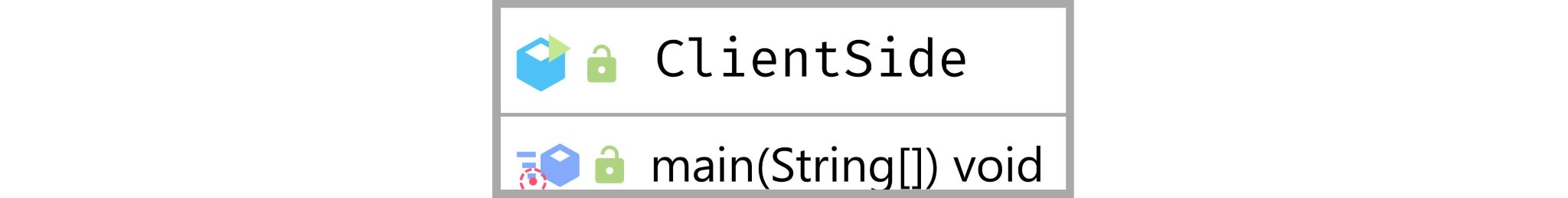
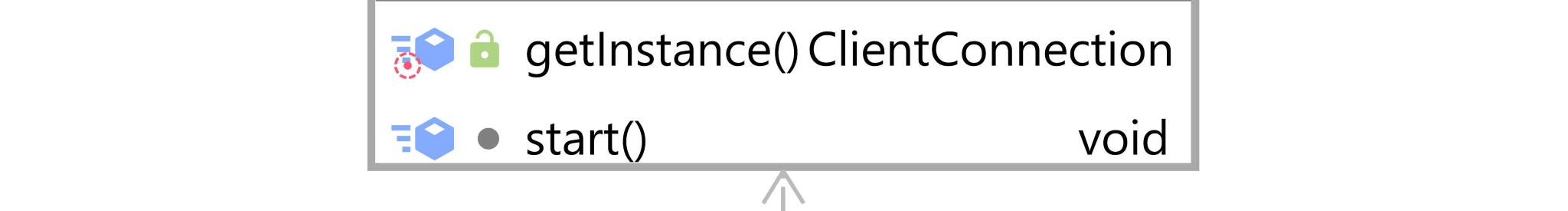
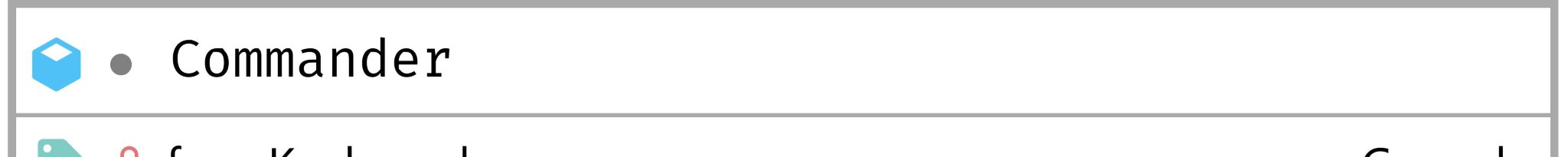
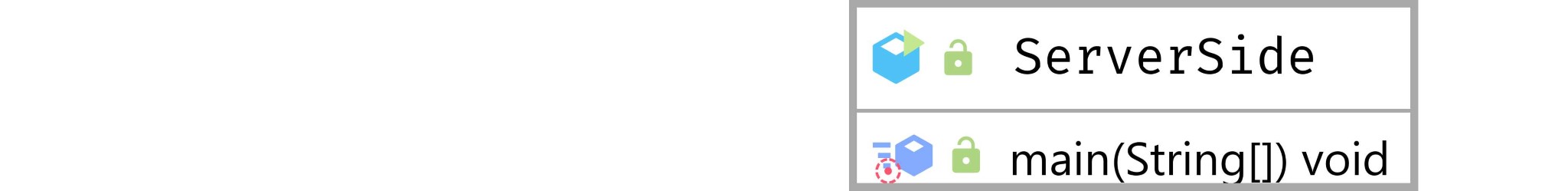
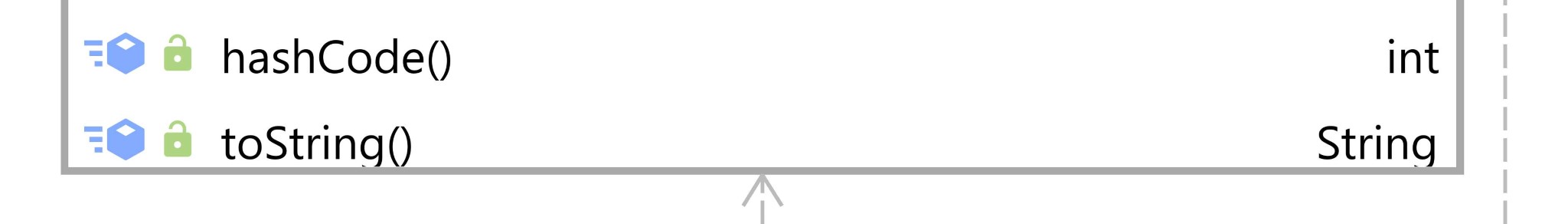
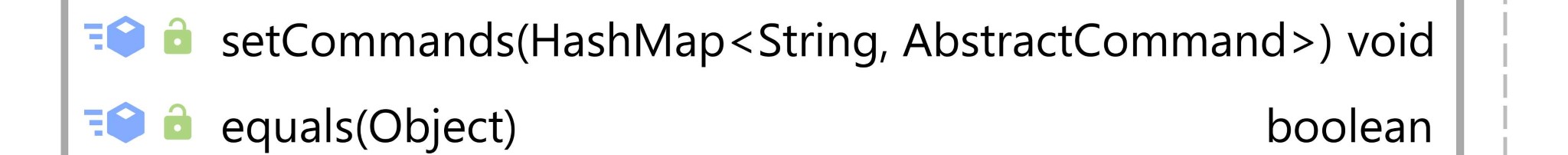
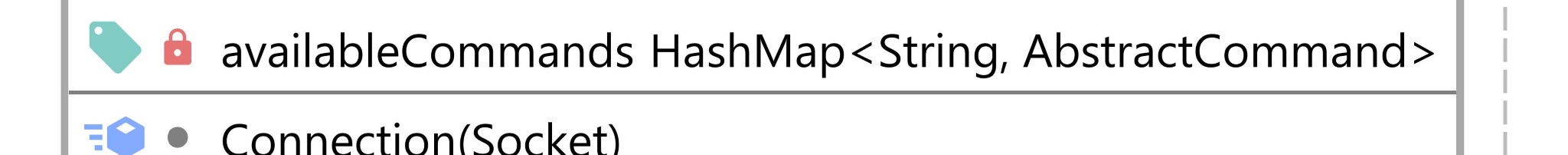
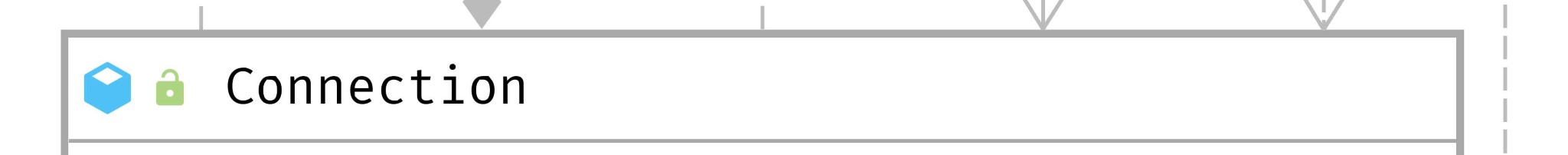
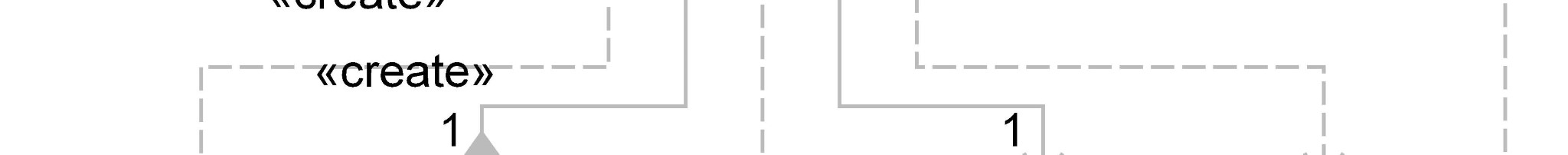
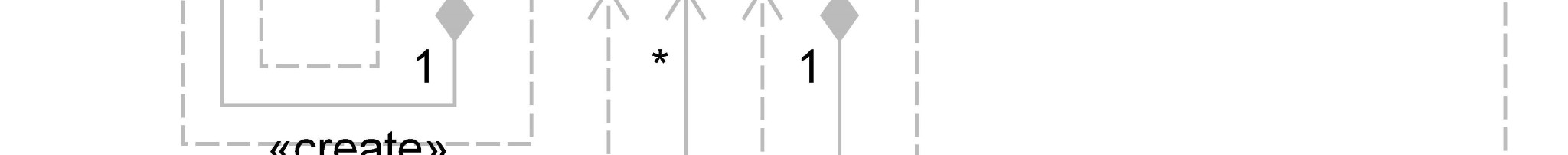
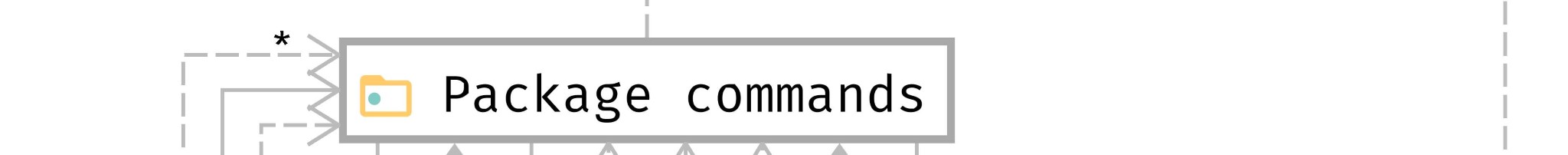
1. В класс, объекты которого хранятся в коллекции, добавить поле типа java.time.OffsetDateTime, в котором должны храниться дата и время создания объекта.
2. Обеспечить возможность регистрации и авторизации пользователей.
3. При регистрации генерировать случайный пароль и отправлять его на почту, указанную при регистрации.
4. Для отправки почтовых уведомлений использовать JavaMail API.
5. Пароли при хранении хэшировать алгоритмом SHA-1.
6. Реализовать ассоциацию между объектом из коллекции и пользователем, его создавшим. Пользователи могут просматривать объекты, созданные всеми пользователями, а модифицировать - только свои.
7. Для идентификации пользователя отправлять логин и пароль с каждым запросом.
8. Обеспечить хранение всех данных (объектов коллекции, пользователей и ассоциаций) в реляционной базе данных

**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.
3. Данные для подключения к почтовому серверу уточняются.

# Диаграмма классов разработанной программы.

## Клиентский модуль: Серверный модуль:



# Выводы по работе.

В процессе выполнения лабораторной работы был дополнен серверный модуль из лабораторной №6 согласно варианту задания. Архитектура разработанного модуля позволила легко расширить проект классами **LoginCommand** и **RegisterCommand** для авторизации и регистрации соответственно. При реализации этих классов возникла проблема в виде **ClassDefNotFoundException** вовремя попытке отправки сообщений используя **Java Mail API**. Classpath и путь к библиотеке были заданы верно, что поставило меня в тупик на некоторое время. Избавиться от проблемы помогла смена JDK12 на JDK8, но первопричину ошибки я так и не установил.

Класс **CollectionManager**, единственный объект которого использовался в программе глобально, и передававшийся по ссылке, был переработан в соответствии с шаблоном Singletone. С учётом полученного опыта был создан класс **DatabaseManager**, предоставляющий доступ к базе данных.

Для отделения кода от прикладных данных было решено использовать файлы **.properties**. Первоначально компилятор отказывался видеть директорию resourses, но установка одноименной метки в среде разработки решило эту проблему.

Последняя сложность заключалась в сохранении временной метки вида **OffsetDateTime** в БД, так как тип данных timestamp в PostgreSQL хранит меньше информации (отсутствуют наносекунды). Решений было 2:

1. Конвертация **OffsetDateTime** в **util.Date**, **sql.Date**, **sql.Timestamp** потерей информации.
2. Сериализация **OffsetDateTime** в json-строку и сохранение в виде текста в БД.

Мной был выбран второй вариант.

Полученные при выполнение лабораторной работы навыки пригодятся при разработке систем, взаимодействующих с реляционными базами данных.