

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
Высшего образования  
*Факультет Программной Инженерии и Компьютерной Техники*

**Лабораторная работа 6 по программированию**

Многопоточность и базы данных

Вариант № 9872

Группа: Р3207

Выполнил:

Садовой Г.В.

Проверил:

Гаврилов А.В.

Г. Санкт-Петербург

2024

# Оглавление

|  |   |
|--|---|
| Текст задания .....                            | 3 |
| Диаграмма классов разработанной программы..... | 5 |
| Клиент .....                                   | 5 |
| Общее .....                                    | 6 |
| Сервер.....                                    | 7 |
| Исходный код программы.....                    | 8 |
| Вывод.....                                     | 9 |

# Текст задания

Внимание! У разных вариантов разный текст задания!

Разделить программу из лабораторной работы №5 на клиентский и серверный модули. Серверный модуль должен осуществлять выполнение команд по управлению коллекцией. Клиентский модуль должен в интерактивном режиме считывать команды, передавать их для выполнения на сервер и выводить результаты выполнения.

Необходимо выполнить следующие требования:

Операции обработки объектов коллекции должны быть реализованы с помощью Stream API с использованием лямбда-выражений.

Объекты между клиентом и сервером должны передаваться в сериализованном виде.

Объекты в коллекции, передаваемой клиенту, должны быть отсортированы по названию

Клиент должен корректно обрабатывать временную недоступность сервера.

Обмен данными между клиентом и сервером должен осуществляться по протоколу UDP

Для обмена данными на сервере необходимо использовать датаграммы

Для обмена данными на клиенте необходимо использовать сетевой канал

Сетевые каналы должны использоваться в неблокирующем режиме.

Обязанности серверного приложения:

Работа с файлом, хранящим коллекцию.

Управление коллекцией объектов.

Назначение автоматически генерируемых полей объектов в коллекции.

Ожидание подключений и запросов от клиента.

Обработка полученных запросов (команд).

Сохранение коллекции в файл при завершении работы приложения.

Сохранение коллекции в файл при исполнении специальной команды, доступной только серверу (клиент такую команду отправить не может).

Серверное приложение должно состоять из следующих модулей (реализованных в виде одного или нескольких классов):

Модуль приёма подключений.

Модуль чтения запроса.

Модуль обработки полученных команд.

Модуль отправки ответов клиенту.

Сервер должен работать в однопоточном режиме.

Обязанности клиентского приложения:

Чтение команд из консоли.

Валидация вводимых данных.

Сериализация введенной команды и её аргументов.

Отправка полученной команды и её аргументов на сервер.

Обработка ответа от сервера (вывод результата исполнения команды в консоль).

Команду save из клиентского приложения необходимо убрать.

Команда exit завершает работу клиентского приложения.

Важно! Команды и их аргументы должны представлять из себя объекты классов. Недопустим обмен "простыми" строками. Так, для команды add или её аналога необходимо сформировать объект, содержащий тип команды и объект, который должен храниться в вашей коллекции.

Дополнительное задание:

Реализовать логирование различных этапов работы сервера (начало работы, получение нового подключения, получение нового запроса, отправка ответа и т.п.) с помощью Log4J2

Отчёт по работе должен содержать:

Текст задания.

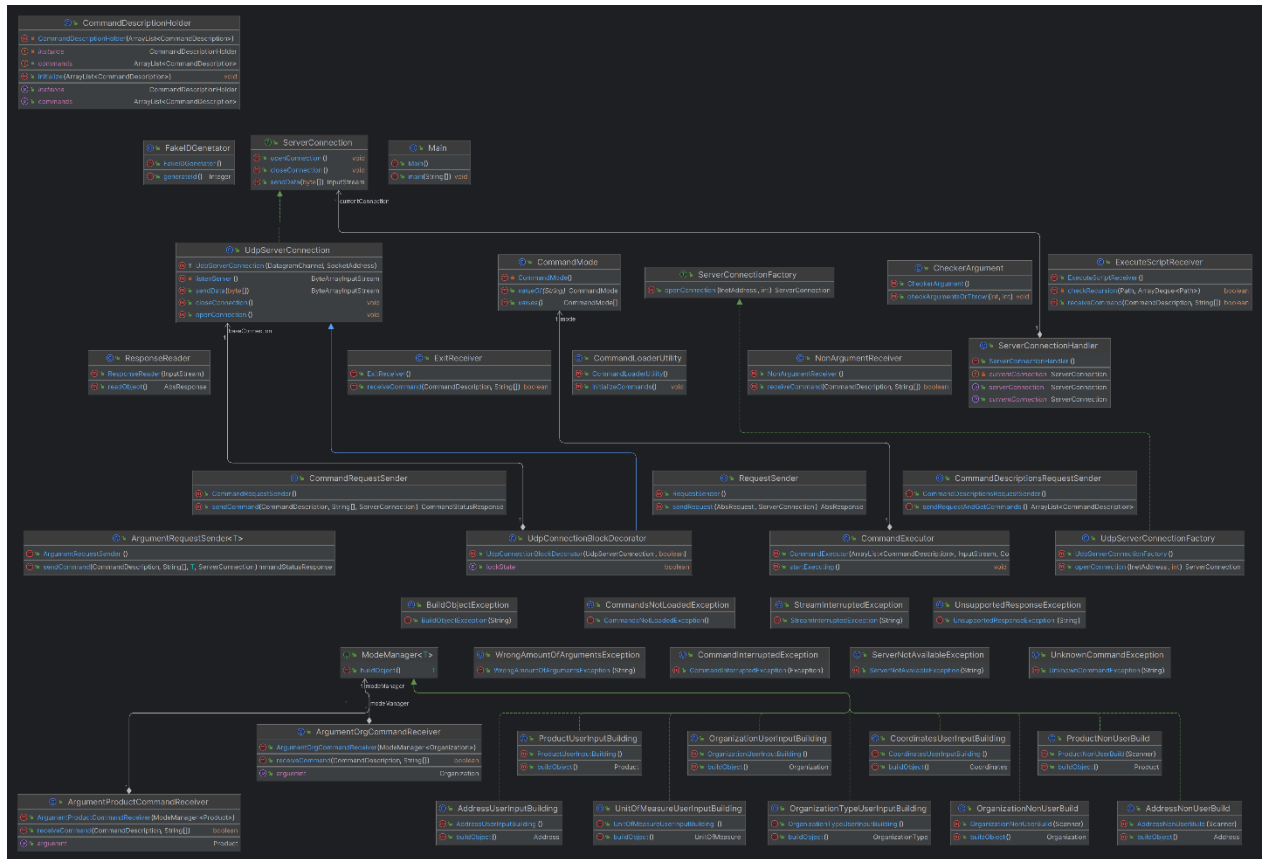
Диаграмма классов разработанной программы (как клиентского, так и серверного приложения).

Исходный код программы.

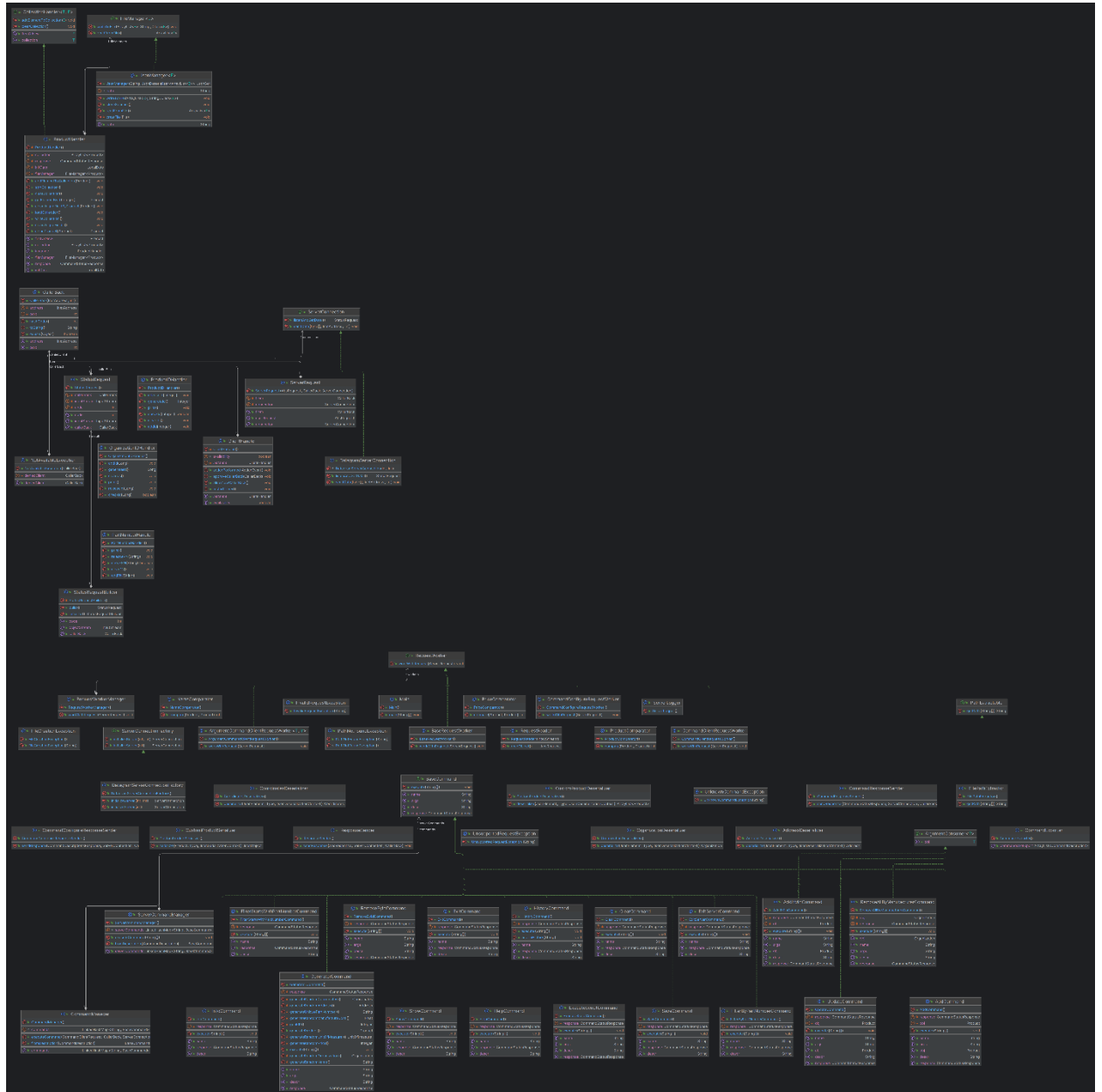
Выводы по работе.

# Диаграмма классов разработанной программы

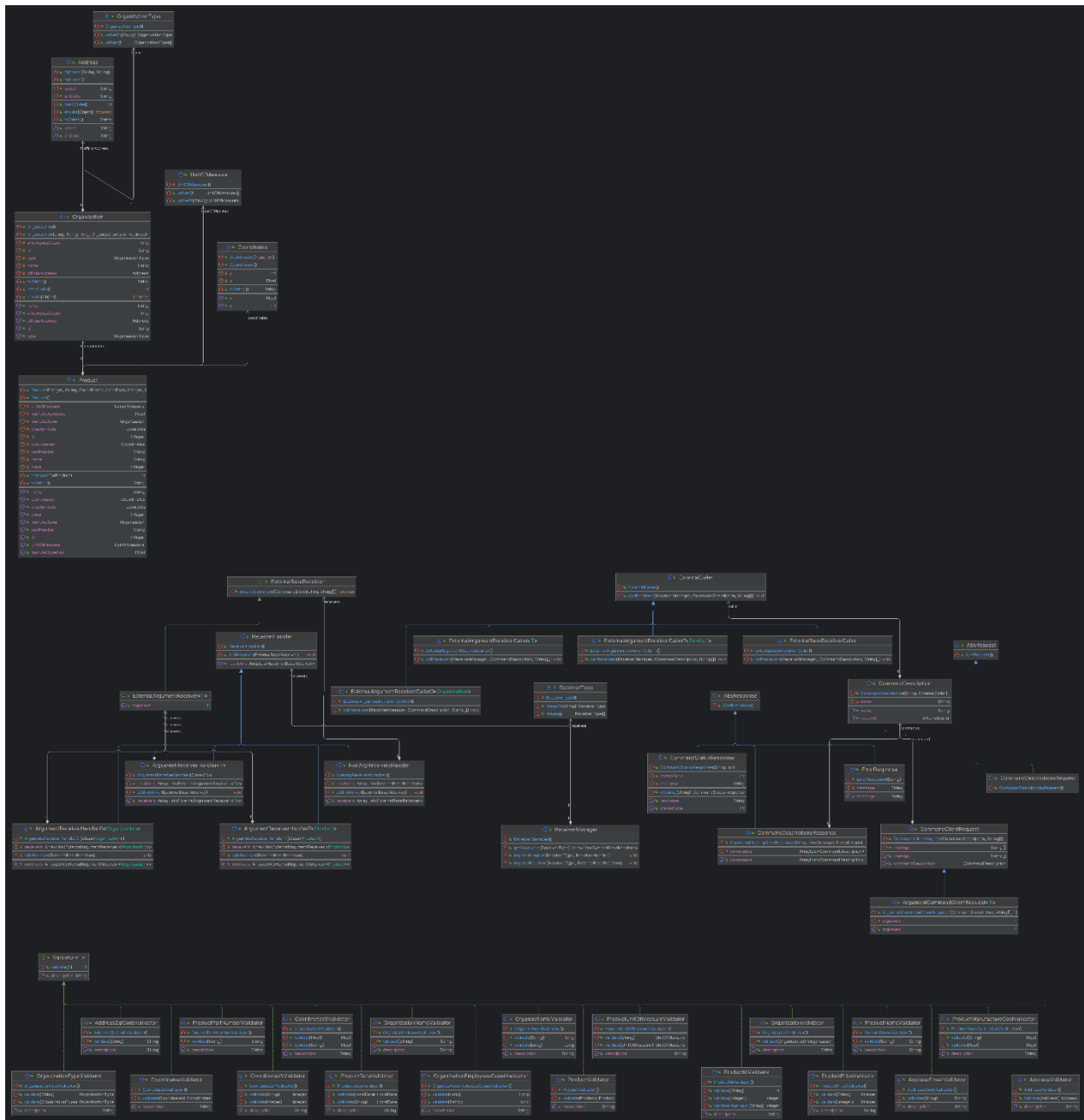
## Клиент



# Общие



## Сервер



# Исходный код программы.

<https://github.com/Umchik1222/MyITMOProjects>



## Вывод

Во время разработки клиент-серверного приложения я познакомился с протоколом UDP. Разработал клиент-серверную архитектуру на основе пятой лабораторной, поделив её на клиент и на сервер. Научился сереализации классов. Углубил знания о логировании.