Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Программирование

**Лабораторная работа № 6**

Вариант № 3113200

Выполнил:

Сандов Кирилл Алекссевич

Группа:

P3113

Проверил:

преподаватель практики Письмак Алексей Евгеньевич

Санкт-Петербург

2023

# Оглавление

[Оглавление 2](#_Toc119848939)

[Задание 3](#_Toc119848940)

[Диаграмма классов модели 5](#_Toc119848941)

[Исходный код программы 6](#_Toc119848942)

[Результат работы программы 7](#_Toc119848943)

[Заключение 10](#_Toc119848944)

# Задание

Разделить программу из [лабораторной работы №5](https://se.ifmo.ru/courses/programming#lab5) 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 TCP
* Для обмена данными на сервере необходимо использовать **сетевой канал**
* Для обмена данными на клиенте необходимо использовать **потоки ввода-вывода**
* 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 введённой 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 **Logback**

# Диаграмма классов модели

## Приложение-сервер

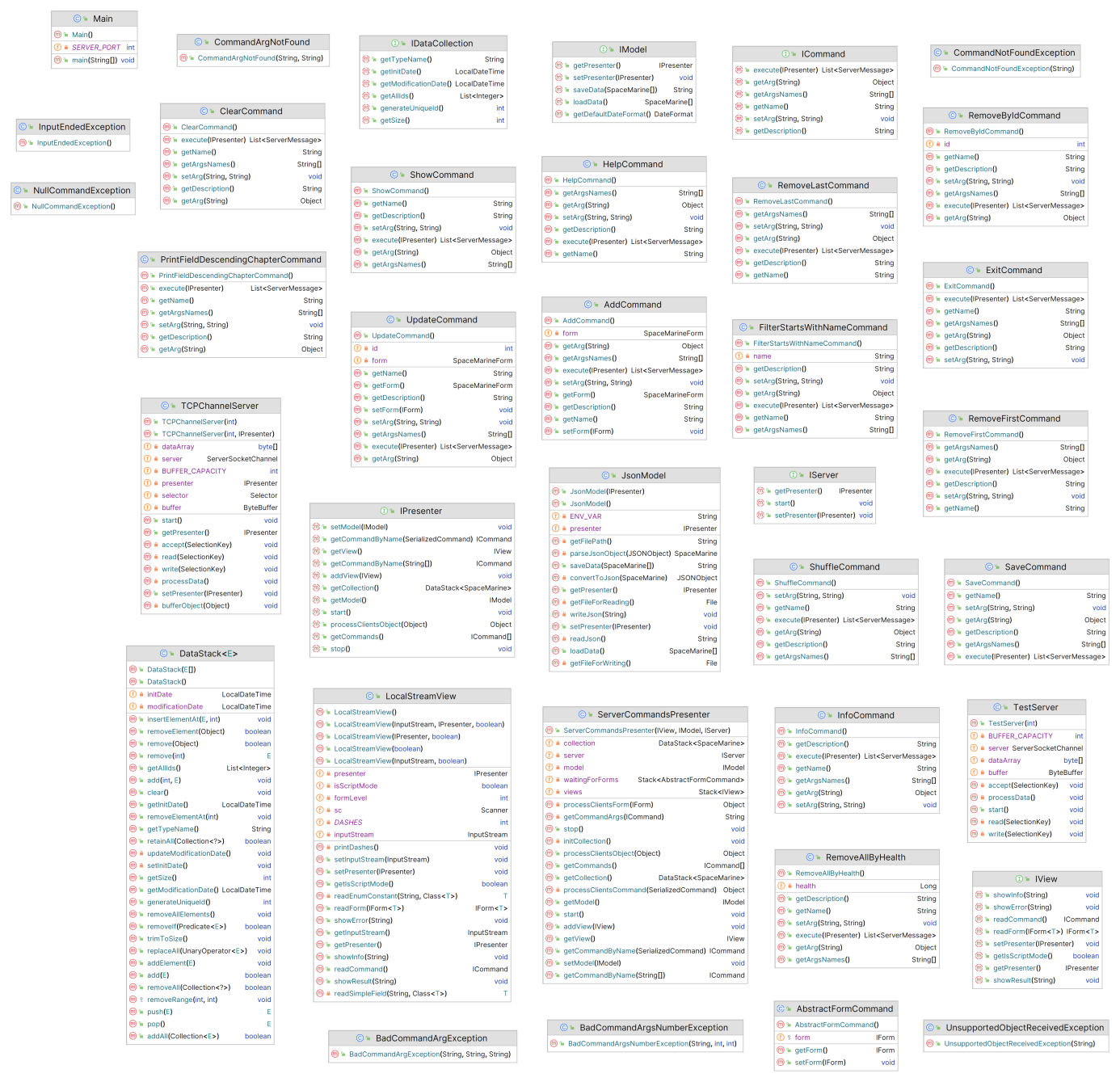


Рисунок 1 – «UML-диаграмма классов сервера»

## Приложение-клиент

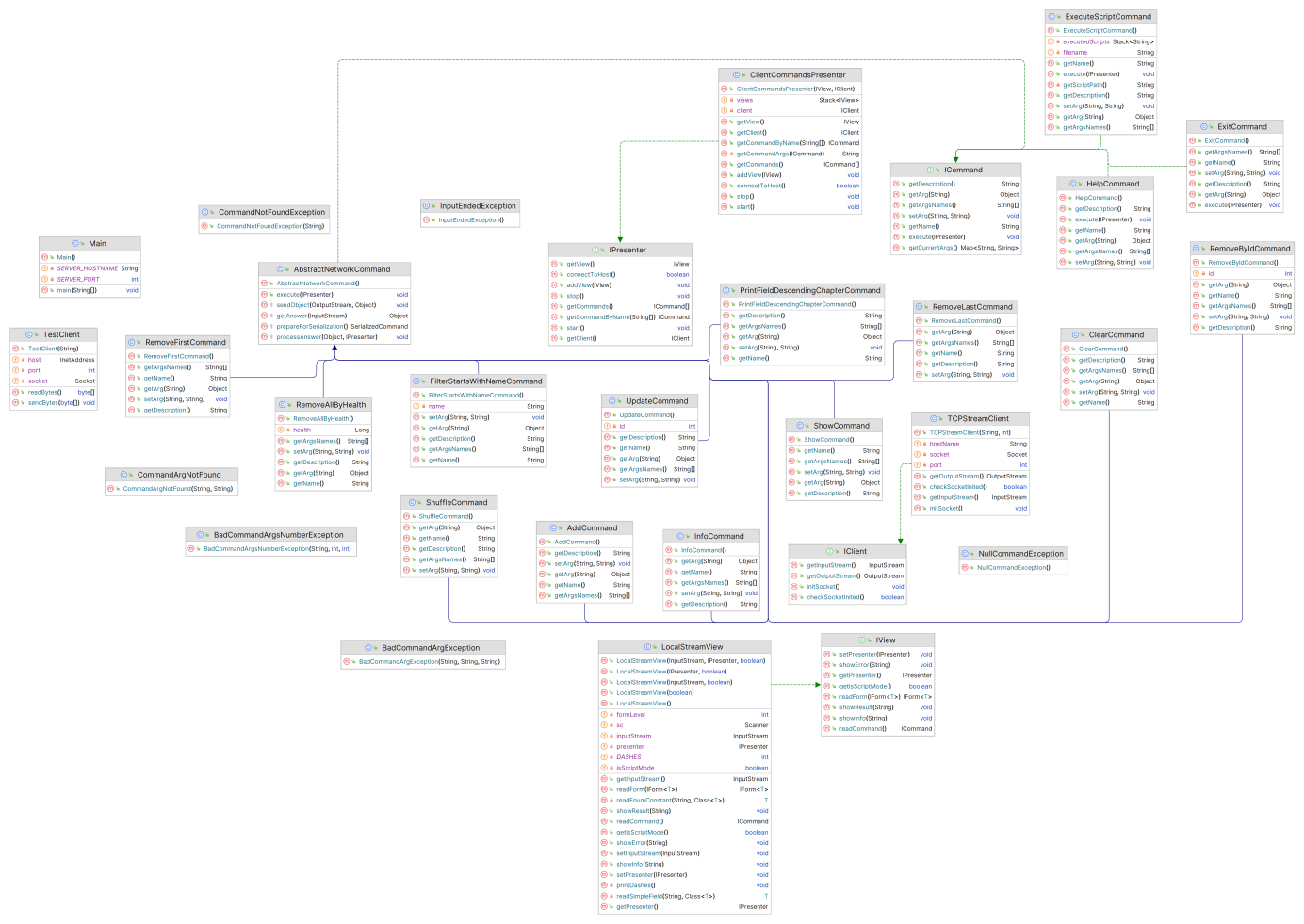


Рисунок 2 – «UML-диаграмма классов клиента»

# Исходный код программы

Ссылка на GitHub-репозиторий с реализацией:

<https://github.com/amphyxs/vt-labas/tree/main/sem-2/prog/lab-6>

# **Заключение**

В результате выполнения данной лабораторной работы изучено сетевое взаимодействие типа «клиент-сервер» и методы его реализации с помощью Java. Во-первых, рассмотрены основные классы из пакета java.net (Socket, SocketChannel, Selector и др.). Получены знания о применения буферов и потоков ввода-вывода для обмена данными. В итоге был реализован полноценный веб-сервер, работающий в неблокирующем режиме, и клиент для него.