Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет

ИТМО»

*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.04 – Системное и прикладное программное обеспечение*

*Дисциплина «Программирование»*

**Отчет**

**По лабораторной работе №6**

**Вариант №1755**

Выполнил:

Кирячек Тимофей Алексеевич

Группа: P3109

Преподаватель:  
Остапенко Иван Вадимович

Г. Санкт-Петербург, 2024 г.

**Оглавление**

[Задание 2](#_Toc158638123)

[UML-диаграмма классов: 5](#_Toc158638124)

[Код программы: 6](#_Toc158638125)

[Заключение 7](#_Toc158638126)

[Список литературы. 8](#_Toc158638127)

# 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 местоположению
* 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.

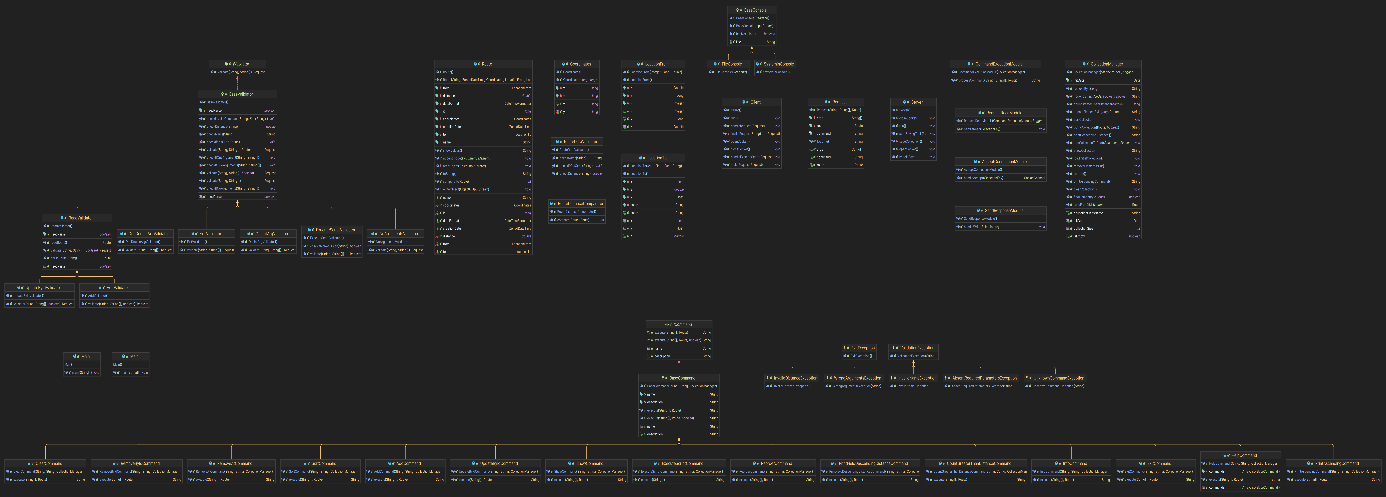
**Обязанности клиентского приложения:**

* 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 **Log4J2**

# UML-диаграмма классов:



# Код программы:

Исходный код может быть найден в [этом репозитории](https://github.com/TecHeReTiC3141/Lab6)

# Заключение

В процессе выполнения лабораторной работы я разработал консольное приложение, получил опыт использования встроенных коллекций, например, Stack, освоил файловый ввод-вывод, узнал о Command pattern и реализовал его, а кроме того, научился создавать JavaDoс и более-менее разумно документировать свой код, без этого такую комплексное приложение было очень тяжело разрабатывать

# Список литературы.

1. https://se.ifmo.ru/courses/programming
2. <https://javarush.ru/groups/posts/1981-dlja-chego-v-java-nuzhnih-interfeysih>
3. <https://javarush.ru/groups/posts/591-vot-tak-final>
4. <https://www.youtube.com/watch?v=3Xo6zSBgdgk>
5. <https://javarush.ru/groups/posts/1985-raznica-mezhdu-abstraktnihmi-klassami-i-interfeysami>
6. <https://skillbox.ru/media/base/klassy-i-obekty-v-java/>
7. <https://www.jetbrains.com/help/idea/class-diagram.html#analyze_graph>
8. <https://metanit.com/java/tutorial/3.3.php>
9. JAVA from EPAM, 2-е издание, исправленное. И. Н. Блинов, В. С. Романчук
10. <https://javarush.ru/groups/posts/1967-otnoshenija-mezhdu-klassami-nasledovanie-kompozicija-i-agregirovanie->
11. <https://javarush.ru/groups/posts/599-inicializacija-poley>
12. https://javarush.ru/groups/posts/2347-klass-string-v-java