

# Министерство образования и науки РФ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»

*Факультет систем управления и робототехники*



## Лабораторная работа №6 По дисциплине : «Программирование» Вариант 665598

Преподаватель: **Максимова Марина Михайловна**

Выполнил: **Нгуен Тоан**

Группа: **R3137**

## Лабораторная работа #6

---

### 1. Текст задания.

#### Требования

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 TCP.
- Для обмена данными на сервере необходимо использовать **поток ввода-вывода**.
- 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 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**

## 2. Диаграмма классов разработанной программы

- Смотрите диаграмму классов сервера [здесь](#)
- Смотрите диаграмму классов клиента [здесь](#)

## 3. Исходный код программы

### Структура папок

Рабочая область по умолчанию содержит три проекта, где:

- **server**: источник сервера
- **client**: источник клиента
- **utilities**: проект предоставляет необходимые объекты как для клиента, так и для сервера

## 4. Результаты программы

### 4.1 Консоль сервера

```
> Task :server:run
14:43:33,921 |-INFO in ch.qos.logback.classic.LoggerContext[default] -
Found resource [logback-test.xml] at
[file:/home/mrtoan/Documents/Java/ProgrammingLab6/server/build/resources/main/logback-test.xml]
14:43:34,012 |-INFO in
ch.qos.logback.classic.joran.action.ConfigurationAction - debug attribute
not set
14:43:34,019 |-INFO in ch.qos.logback.core.joran.action.AppenderAction -
About to instantiate appender of type [ch.qos.logback.core.ConsoleAppender]
14:43:34,021 |-INFO in ch.qos.logback.core.joran.action.AppenderAction -
Naming appender as [consoleAppender]
14:43:34,026 |-INFO in
ch.qos.logback.core.joran.action.NestedComplexPropertyIA - Assuming default
type [ch.qos.logback.classic.encoder.PatternLayoutEncoder] for [encoder]
property
14:43:34,071 |-INFO in ch.qos.logback.core.joran.action.AppenderAction -
About to instantiate appender of type
[ch.qos.logback.core.rolling.RollingFileAppender]
14:43:34,076 |-INFO in ch.qos.logback.core.joran.action.AppenderAction -
Naming appender as [fileAppender]
14:43:34,089 |-INFO in c.q.l.core.rolling.TimeBasedRollingPolicy@1928931046
- No compression will be used
14:43:34,090 |-INFO in c.q.l.core.rolling.TimeBasedRollingPolicy@1928931046
- Will use the pattern ../logs/application_%d{yyyy-MM-dd}.%i.log for the
active file
14:43:34,092 |-INFO in
ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP@3daa422a - The date
pattern is 'yyyy-MM-dd' from file name pattern
'../logs/application_%d{yyyy-MM-dd}.%i.log'.
14:43:34,092 |-INFO in
ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP@3daa422a - Roll-over at
midnight.
14:43:34,097 |-INFO in
ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP@3daa422a - Setting
initial period to Mon Apr 25 14:42:22 MSK 2022
14:43:34,097 |-WARN in
ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP@3daa422a -
SizeAndTimeBasedFNATP is deprecated. Use SizeAndTimeBasedRollingPolicy
instead
14:43:34,097 |-WARN in
ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP@3daa422a - For more
information see
http://logback.qos.ch/manual/appenders.html#SizeAndTimeBasedRollingPolicy
```

```
14:43:34,099 |-INFO in
ch.qos.logback.core.joran.action.NestedComplexPropertyIA - Assuming default
type [ch.qos.logback.classic.encoder.PatternLayoutEncoder] for [encoder]
property
14:43:34,101 |-INFO in
ch.qos.logback.core.rolling.RollingFileAppender[fileAppender] - Active log
file name: ../logs/application.log
14:43:34,101 |-INFO in
ch.qos.logback.core.rolling.RollingFileAppender[fileAppender] - File
property is set to [../logs/application.log]
14:43:34,102 |-INFO in ch.qos.logback.classic.joran.action.RootLoggerAction
- Setting level of ROOT logger to DEBUG
14:43:34,102 |-INFO in ch.qos.logback.core.joran.action.AppenderRefAction -
Attaching appender named [consoleAppender] to Logger[ROOT]
14:43:34,102 |-INFO in ch.qos.logback.core.joran.action.AppenderRefAction -
Attaching appender named [fileAppender] to Logger[ROOT]
14:43:34,102 |-INFO in
ch.qos.logback.classic.joran.action.ConfigurationAction - End of
configuration.
14:43:34,102 |-INFO in
ch.qos.logback.classic.joran.JoranConfigurator@31c88ec8 - Registering
current configuration as safe fallback point
```

Collection loaded successfully.

WARNING: An illegal reflective access operation has occurred

WARNING: Illegal reflective access by

com.google.gson.internal.reflect.ReflectionHelper

(file:/home/mrtoan/.gradle/caches/modules-2/files-

2.1/com.google.code.gson/gson/2.9.0/8a1167e089096758b49f9b34066ef98b2f4b37a  
a/gson-2.9.0.jar) to field java.time.LocalDateTime.date

WARNING: Please consider reporting this to the maintainers of

com.google.gson.internal.reflect.ReflectionHelper

WARNING: Use --illegal-access=warn to enable warnings of further illegal  
reflective access operations

WARNING: All illegal access operations will be denied in a future release

INFO | 25-04-2022 14:43:34 | [main] ServerLogger s.u.CollectionFileManager  
- Collection loaded successfully.

INFO | 25-04-2022 14:43:34 | [main] ServerLogger s.Server - Server  
start...

INFO | 25-04-2022 14:43:34 | [main] ServerLogger s.Server - The server  
started successfully.

Port listening '4999'...

INFO | 25-04-2022 14:43:34 | [main] ServerLogger s.Server - Port listening  
'4999'...

The connection with the client was successfully established.

INFO | 25-04-2022 14:44:24 | [main] ServerLogger s.Server - The connection  
with the client was successfully established.

INFO | 25-04-2022 14:47:11 | [main] ServerLogger s.Server - Request 'show'  
processed successfully.

INFO | 25-04-2022 14:47:30 | [main] ServerLogger s.Server - Request 'help'  
processed successfully.

INFO | 25-04-2022 14:47:43 | [main] ServerLogger s.Server - Request  
'average\_of\_annual\_turnover' processed successfully.

INFO | 25-04-2022 14:48:28 | [main] ServerLogger s.Server - Request

```
'filter_greater_than_type' processed successfully.
INFO | 25-04-2022 14:48:39 | [main] ServerLogger s.Server - Request 'exit'
processed successfully.
Client successfully disconnected from server!
INFO | 25-04-2022 14:48:39 | [main] ServerLogger s.Server - Client
successfully disconnected from server!
Port listening '4999'...
INFO | 25-04-2022 14:48:39 | [main] ServerLogger s.Server - Port listening
'4999'...
error: Connection timed out!
WARN | 25-04-2022 14:49:39 | [main] ServerLogger s.Server - Connection
timed out!
error: Connection timed out. Reconnect!
ERROR | 25-04-2022 14:49:39 | [main] ServerLogger s.Server - Connection
timed out. Reconnect!
INFO | 25-04-2022 14:49:39 | [main] ServerLogger s.Server - Shutting down
the server...
Server operation completed successfully.
INFO | 25-04-2022 14:49:39 | [main] ServerLogger s.Server - Server
operation completed successfully.
```

## 4.2 Консоль Клиента

```
> Task :client:run
The connection to the server was successfully established.
Waiting for permission to share data...
Permission to share data has been received.
$ show
Organization{
    id=1,
    name='Thuyenf',
    coordinates=Coordinates{x=12, y=21.0},
    creationDate=2022-04-20T00:36:01.718718,
    annualTurnover=1000,
    organizationType=PUBLIC,
    officialAddress=Address{street='nguyuentoan', zipCode='1000dd'}}

Organization{
    id=2,
    name='toandada',
    coordinates=Coordinates{x=12, y=12.0},
    creationDate=2022-04-20T21:58:38.375277,
    annualTurnover=1000,
    organizationType=PUBLIC,
    officialAddress=Address{street='null', zipCode='121231n31nn'}}

Organization{
    id=3,
    name='toandada',
    coordinates=Coordinates{x=12, y=12.0},
    creationDate=2022-04-20T22:19:09.219322,
```

```

    annualTurnover=-1,
    organizationType=PRIVATE_LIMITED_COMPANY,
    officialAddress=Address{street='null', zipCode='121231n31nn'}}

Organization{
    id=4,
    name='toandada',
    coordinates=Coordinates{x=12, y=12.0},
    creationDate=2022-04-20T22:38:50.008429,
    annualTurnover=1000,
    organizationType=PUBLIC,
    officialAddress=Address{street='null', zipCode='121231n31nn'}}
$ help
help                display help on available commands
info                print information about the collection
to standard output (type, initialization date, number of elements, etc.)
show                display all items in the collection
add {element}        add a new element to the collection
update <ID>          update the value of the collection
element whose id is equal to the given one
remove_by_id <ID>    remove item from collection by ID
clear                clear the collection
save                 save collection to file
exit                 terminate program (without saving to
file)
execute_script <file_name> execute script from specified file
add_if_min {element} update the value of the collection
element whose id is equal to the given one
remove_lower {element} remove from the collection all
elements greater than the specified
history              display history of used commands
average_of_annual_turnover read and execute the script from the
specified file
count_greater_than_official_address {element} print the number of elements
whose officialAddress field value is greater than the specified one
filter_greater_than_type <OrganizationType> display elements whose
organization type field value is equal to the given one
$ average_of_annual_turnover
Average value of annual turnover for all organizations: 749.75
$ filter_greater_than_type government
Organization{
    id=3,
    name='toandada',
    coordinates=Coordinates{x=12, y=12.0},
    creationDate=2022-04-20T22:19:09.219322,
    annualTurnover=-1,
    organizationType=PRIVATE_LIMITED_COMPANY,
    officialAddress=Address{street='null', zipCode='121231n31nn'}}
$ exit
Collection autosaved to file.
The client's work has been completed successfully.

```

## 4.3 Содержимое лог-файла

```
INFO | 25-04-2022 14:43:34 | [main] ServerLogger s.u.CollectionFileManager  
- Collection loaded successfully.  
INFO | 25-04-2022 14:43:34 | [main] ServerLogger s.Server - Server  
start...  
INFO | 25-04-2022 14:43:34 | [main] ServerLogger s.Server - The server  
started successfully.  
INFO | 25-04-2022 14:43:34 | [main] ServerLogger s.Server - Port listening  
'4999'...  
INFO | 25-04-2022 14:44:24 | [main] ServerLogger s.Server - The connection  
with the client was successfully established.  
INFO | 25-04-2022 14:47:11 | [main] ServerLogger s.Server - Request 'show'  
processed successfully.  
INFO | 25-04-2022 14:47:30 | [main] ServerLogger s.Server - Request 'help'  
processed successfully.  
INFO | 25-04-2022 14:47:43 | [main] ServerLogger s.Server - Request  
'average_of_annual_turnover' processed successfully.  
INFO | 25-04-2022 14:48:28 | [main] ServerLogger s.Server - Request  
'filter_greater_than_type' processed successfully.  
INFO | 25-04-2022 14:48:39 | [main] ServerLogger s.Server - Request 'exit'  
processed successfully.  
INFO | 25-04-2022 14:48:39 | [main] ServerLogger s.Server - Client  
successfully disconnected from server!  
INFO | 25-04-2022 14:48:39 | [main] ServerLogger s.Server - Port listening  
'4999'...  
WARN | 25-04-2022 14:49:39 | [main] ServerLogger s.Server - Connection  
timed out!  
ERROR | 25-04-2022 14:49:39 | [main] ServerLogger s.Server - Connection  
timed out. Reconnect!  
INFO | 25-04-2022 14:49:39 | [main] ServerLogger s.Server - Shutting down  
the server...  
INFO | 25-04-2022 14:49:39 | [main] ServerLogger s.Server - Server  
operation completed successfully.
```

## 5. Выводы по работе

Во время выполнения данной лабораторной работы мы закрепили принципы SOLID, собственные исключения и многое другое. Также мы научились использовать Javadoc, работать с потоками, файлами, интерфейсами Comparable и Comparator. Узнали что такое сериализация и десериализация.