



**ITMO UNIVERSITY**

**Лабораторная работа №6**

**по программированию**

**Вариант 289343**

**Выполнил:**

Пурэвсурэн Билгуун

Группа Р3113

**Преподаватель:**

**Письмак Алексей Евгеньевич**

# 1. Текст заданий

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 местоположению
- 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 **сетевой канал**
- Для обмена данными на клиенте необходимо использовать **датаграммы**
- 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.

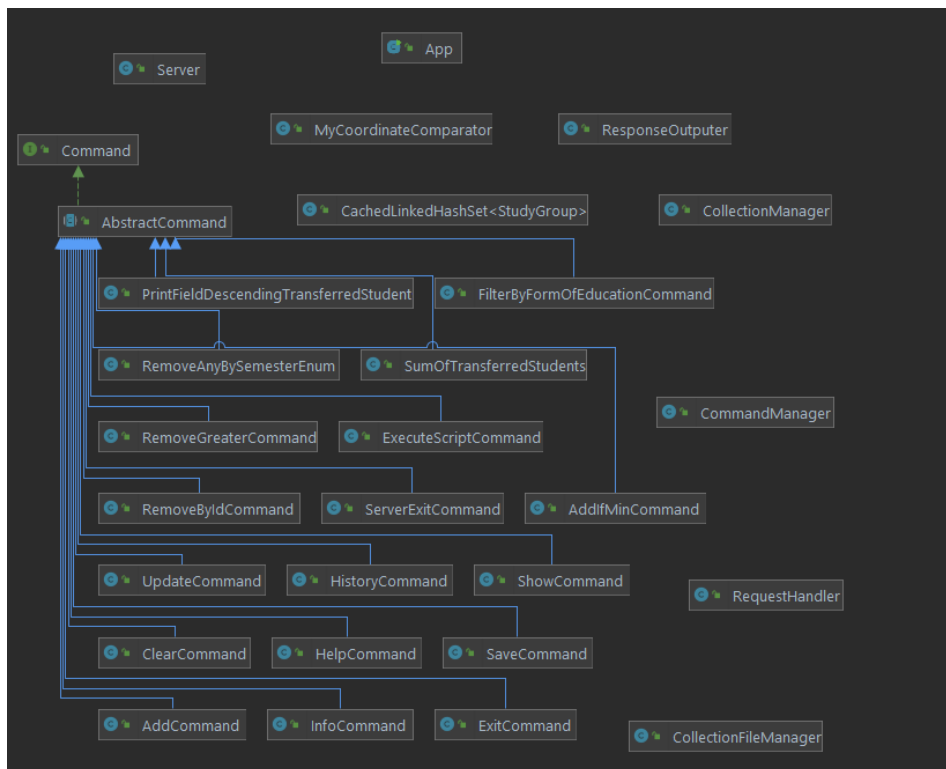
**Важно!** Команды и их аргументы должны представлять из себя объекты классов. Недопустим обмен "простыми" строками. Так, для команды add или её аналога необходимо сформировать объект, содержащий тип команды и объект, который должен храниться в вашей коллекции.

**Дополнительное задание:**

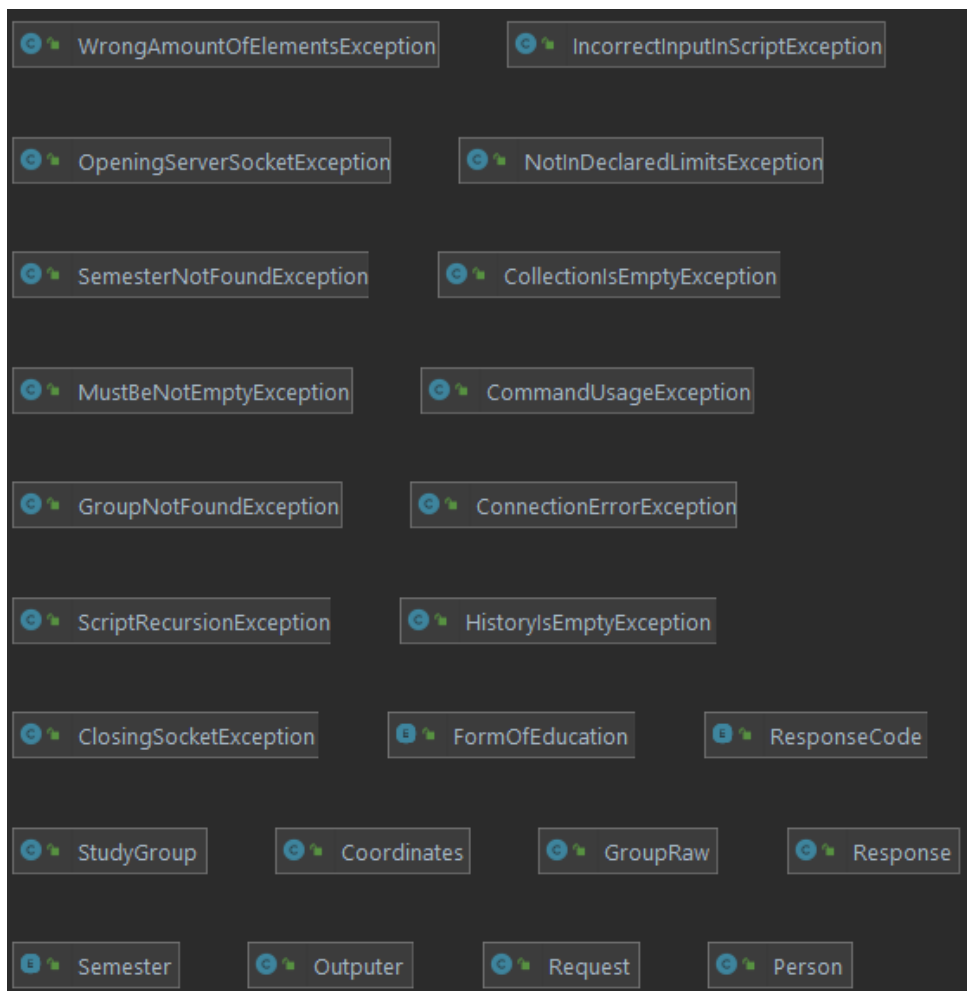
Реализовать логирование различных этапов работы сервера (начало работы, получение нового подключения, получение нового запроса, отправка ответа и т.п.) с помощью **Logback**

## 2. Диаграмма классов разработанной программы

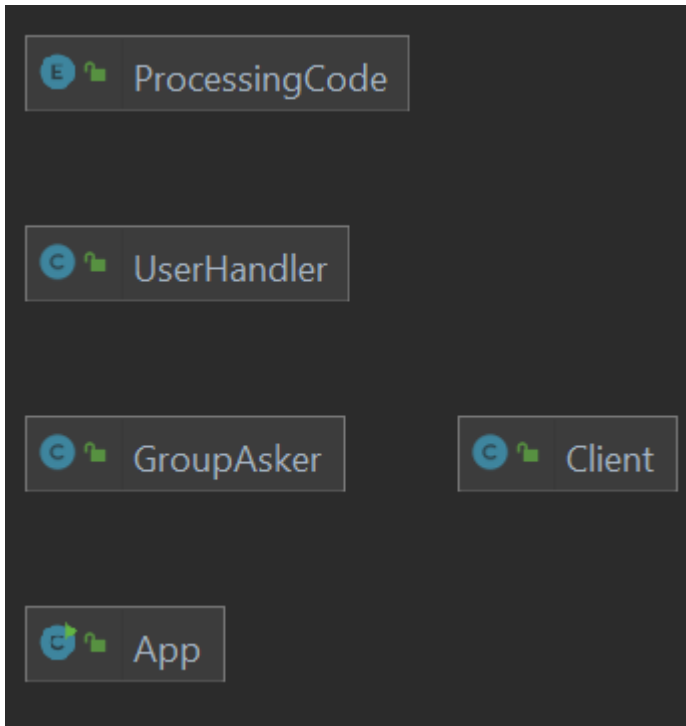
### Server



### Common



## Client



## 3. Исходный код программы

Ссылка на гитхаб:

<https://github.com/bilguuk124/Programming-Labs/tree/master/Lab6>

## 4. Вывод программы

### Help Command:

Client:

```
$ help
I've sent the package
I've received a package
help          вывести справку по доступным командам
info          вывести информацию о коллекции
show          вывести все элементы коллекции
add           добавить новый элемент в коллекцию
update        обновить значение элемента коллекции по ID
remove_by_id  удалить элемент из коллекции по ID
clear         очистить коллекцию
save          сохранить коллекцию в файл
exit          завершить программу (без сохранения в файл)
execute_script исполнить скрипт из указанного файла
add_if_min    добавить новый элемент, если его значение меньше
remove_greater удалить из коллекции все элементы, превышающие
history       вывести историю использованных команд
sum_of_transferred_students вывести сумму переведённых студентов для всех
remove_by_semester удалить из коллекции один элемент, значение
filter_by_education_form вывести элементы, значение поля educationForm
server_exit   завершить работу сервера
print_descending_transferred вывести значения поля transferredStudents в
```

Server:

```
Коллекция успешно загружена
22:52:26.432 [main] INFO server - Коллекция успешно загружена
22:52:26.518 [main] INFO server - Запуск сервера...
22:52:26.914 [main] INFO server - Сервер успешно запущен.
22:53:14.615 [main] INFO server - Получено команда 'help'
22:53:14.622 [main] INFO server - Команда 'help' выполнена и отправлена
|
```

Show Command:

Client:

```
$ show
I've sent the package
I've received a package
Группа № 4 (добавлен Fri May 21 22:28:58 ULAT 20211621607338000)
Имя: The Fourth One
Местоположение: X:0Y: 0.0
Число студентов: 20
Студенты перевелись: 35
Форма обучения: DISTANCE_EDUCATION
Семестр: THIRD
Админ группы: JINKEN ( родился в 2000-01-01, № Паспорта: ok12121212 )

Группа № 3 (добавлен Fri May 21 22:28:16 ULAT 20211621607296000)
Имя: The Third One
Местоположение: X:0Y: 1.0
Число студентов: 25
Студенты перевелись: 0
Форма обучения: FULL_TIME_EDUCATION
Семестр: SIXTH
Админ группы: JINKEN ( родился в 2001-10-25, № Паспорта: OK10010 )
```

Server:

```
22:55:09.519 [main] INFO server - Получено команда 'show'
22:55:09.521 [main] INFO server - Команда 'show' выполнена и отправлена
```

Server\_exit Command:

Client:

```
$ server_exit
I've sent the package
I've received a package
```

## Server:

```
22:56:45.342 [main] INFO server - Команда 'server_exit' выполнена и отправлена
22:56:45.342 [main] INFO server - Завершение работы сервера...
Коллекция успешно сохранена в файл.

22:56:45.348 [main] INFO server - Коллекция сохранена!
22:56:45.348 [main] INFO server - Работа сервера успешно завершена

Process finished with exit code 0
|
```