

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Санкт-Петербургский национальный исследовательский университет  
ИТМО»

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

**ЛАБОРАТОРНАЯ РАБОТА №6**

по дисциплине  
‘ПРОГРАММИРОВАНИЕ’

Вариант №3131257

*Выполнил:*  
Студент группы Р3131  
Нодири Хисравхон

*Преподаватель:*  
Письмак Алексей Евгеньевич



Санкт-Петербург, 2023

Задание:

```
< > readme.md x
1 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.
2
3 Необходимо выполнить следующие требования:
4
5 +Операции обработки объектов коллекции должны быть реализованы с помощью Stream API с использованием лямбда-выражений.
6 +Объекты между клиентом и сервером должны передаваться в сериализованном виде.
7 +Объекты в коллекции, передаваемой клиенту, должны быть отсортированы по местоположению
8 +Клиент должен корректно обрабатывать временную недоступность сервера.
9 +Обмен данными между клиентом и сервером должен осуществляться по протоколу UDP
10 +Для обмена данными на сервере необходимо использовать датаграммы
11 +Для обмена данными на клиенте необходимо использовать сетевой канал
12 +Сетевые каналы должны использоваться в неблокирующем режиме.
13
14 Обязанности серверного приложения:
15
16 +Работа с файлом, хранящим коллекцию.
17 +Управление коллекцией объектов.
18 +Назначение автоматически генерируемых полей объектов в коллекции.
19 +Ожидание подключений и запросов от клиента.
20 +Обработка полученных запросов (команд).
21 +Сохранение коллекции в файл при завершении работы приложения.
22 +Сохранение коллекции в файл при исполнении специальной команды, доступной только серверу (клиент такую команду отправить не может).
23
24 Серверное приложение должно состоять из следующих модулей (реализованных в виде одного или нескольких классов):
25
26 +Модуль приёма подключений.
27 +Модуль чтения запроса.
28 +Модуль обработки полученных команд.
29 +Модуль отправки ответов клиенту.
30
31 Сервер должен работать в однопоточном режиме.
32
33 Обязанности клиентского приложения:
34
35 +Чтение команд из консоли.
36 +Валидация вводимых данных.
37 +Сериализация введённой команды и её аргументов.
38 +Отправка полученной команды и её аргументов на сервер.
39 +Обработка ответа от сервера (вывод результата исполнения команды в консоль).
40
41 +Обработка ответа от сервера (вывод результата исполнения команды в консоль).
```

```
< > readme.md x
29 +Модуль отправки ответов клиенту.
30
31 Сервер должен работать в однопоточном режиме.
32
33 Обязанности клиентского приложения:
34
35 +Чтение команд из консоли.
36 +Валидация вводимых данных.
37 +Сериализация введённой команды и её аргументов.
38 +Отправка полученной команды и её аргументов на сервер.
39 +Обработка ответа от сервера (вывод результата исполнения команды в консоль).
40 +Команду save из клиентского приложения необходимо убрать.
41 +Команда exit завершает работу клиентского приложения.
42
43 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.
44
45 Дополнительное задание:
46 -Реализовать логирование различных этапов работы сервера (начало работы, получение нового подключения, получение нового запроса, отправка ответа и т.п.)
с помощью Java Util Logging
47
48 Отчёт по работе должен содержать:
49
50 Текст задания.
51 Диаграмма классов разработанной программы (как клиентского, так и серверного приложения).
52 Исходный код программы.
53 Выводы по работе.
54
55 Вопросы к защите лабораторной работы:
56
57 Сетевое взаимодействие - клиент-серверная архитектура, основные протоколы, их сходства и отличия.
58 Протокол TCP. Классы Socket и ServerSocket.
59 Протокол UDP. Классы DatagramSocket и DatagramPacket.
60 Отличия блокирующего и неблокирующего ввода-вывода, их преимущества и недостатки. Работа с сетевыми каналами.
61 Классы SocketChannel и DatagramChannel.
62 Передача данных по сети. Сериализация объектов.
63 Интерфейс Serializable. Объектный граф, сериализация и десериализация полей и методов.
64 Java Stream API. Создание конвейеров. Промежуточные и терминальные операции.
65 Шаблоны проектирования: Decorator, Iterator, Factory method, Command, Flyweight, Interpreter, Singleton, Strategy, Adapter, Facade, Proxy.
66
```

**Исходный код доступен по ссылке**

<https://github.com/Kuchizu/ltmoLabs/tree/main/%D0%A1%D0%B5%D0%BC%202/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/Lab%206>

**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.