Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

Высшего образования

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №6 по программированию

Язык Java

Вариант №3451

Группа: P3132

Выполнил:

Гаврилин О. С.

Преподаватель:

Абузов Я.А.

Санкт-Петербург

2024

Содержание

Текст задания

Внимание! У разных вариантов разный текст задания!

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 имени
- 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:

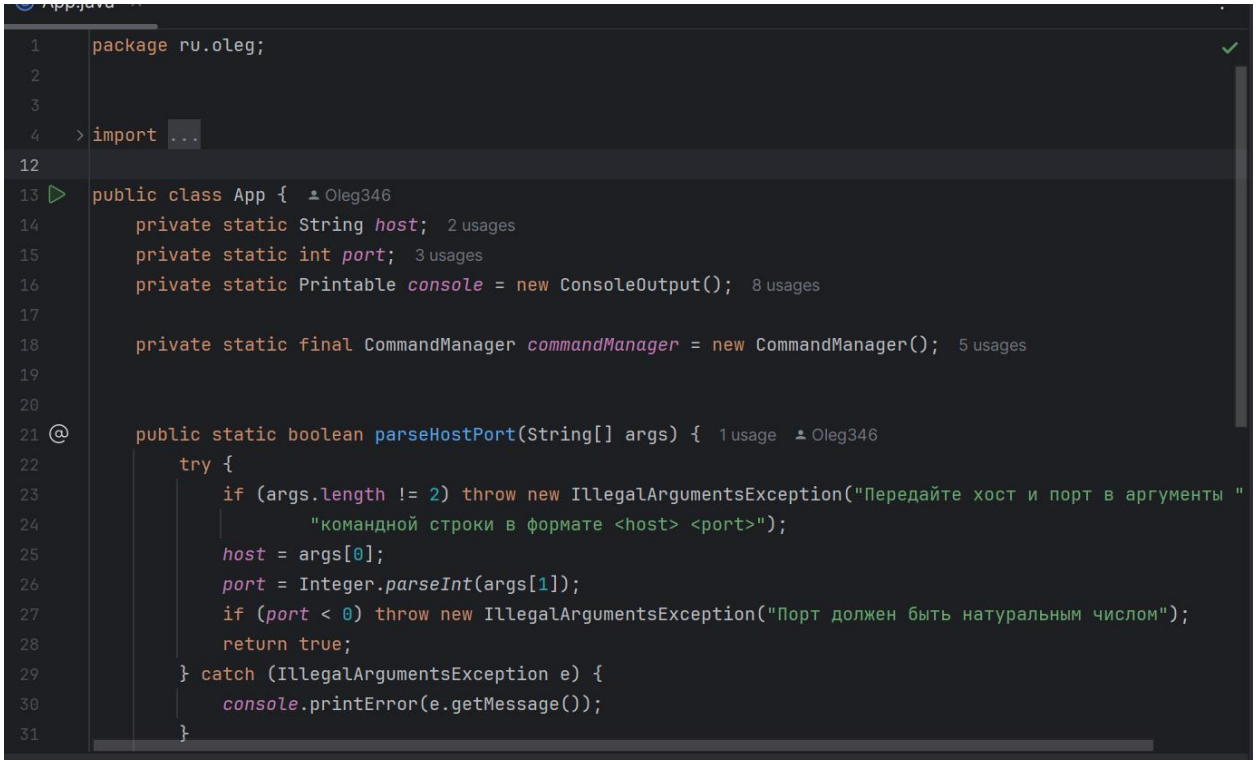
- Чтение команд из консоли.
- Валидация вводимых данных.
- Сериализация введённой команды и её аргументов.
- Отправка полученной команды и её аргументов на сервер.
- Обработка ответа от сервера (вывод результата исполнения команды в консоль).
- Команду `save` из клиентского приложения необходимо убрать.
- 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**

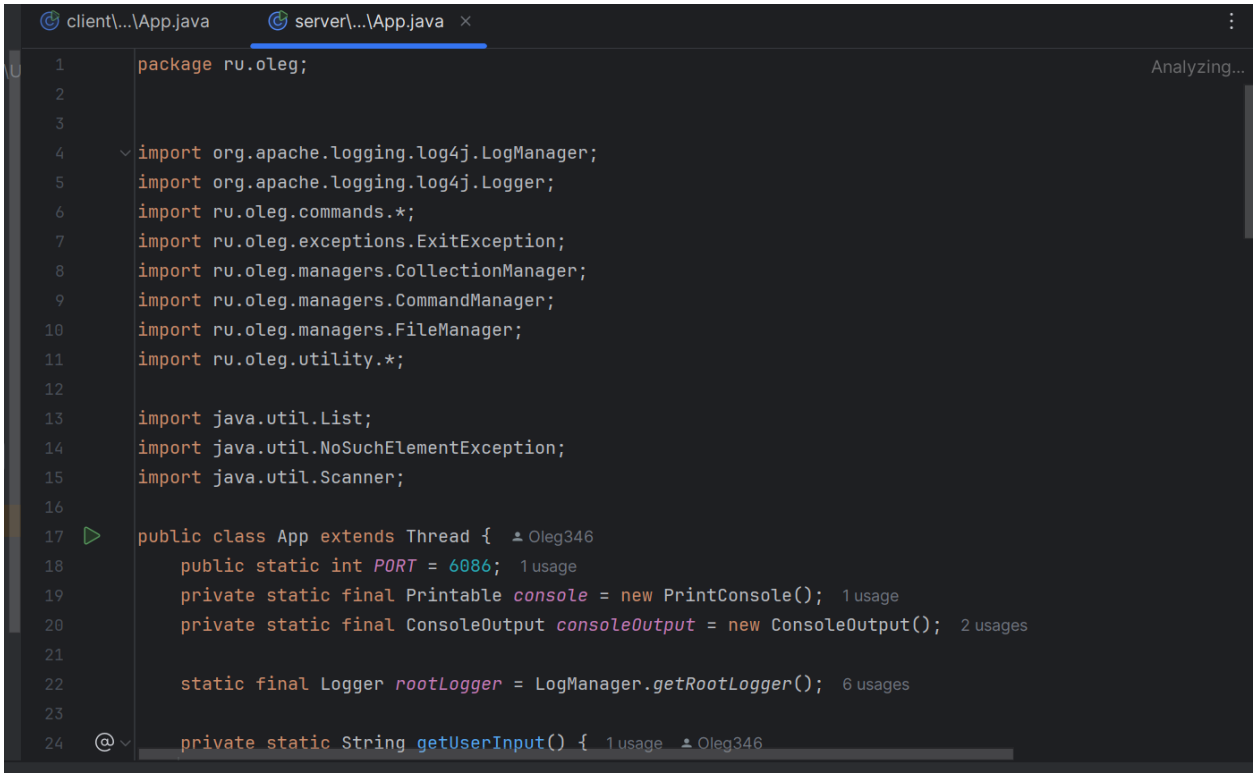
Исходный код.

Основной файл клиента:

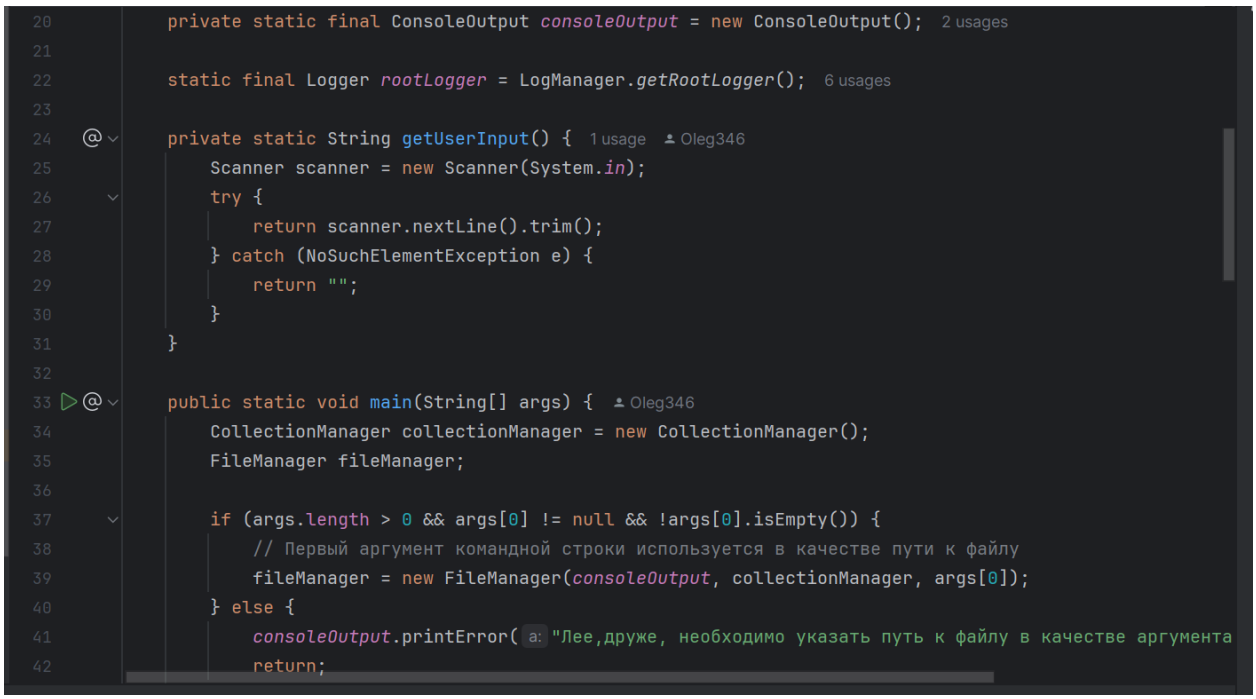


```java
package ru.oleg;


import ...

public class App {  Oleg346
    private static String host;  2 usages
    private static int port;  3 usages
    private static Printable console = new ConsoleOutput();  8 usages

    private static final CommandManager commandManager = new CommandManager();  5 usages


    public static boolean parseHostPort(String[] args) {  1 usage  Oleg346
        try {
            if (args.length != 2) throw new IllegalArgumentsException("Передайте хост и порт в аргументы "
                    "командной строки в формате <host> <port>");
            host = args[0];
            port = Integer.parseInt(args[1]);
            if (port < 0) throw new IllegalArgumentsException("Порт должен быть натуральным числом");
            return true;
        } catch (IllegalArgumentsException e) {
            console.printError(e.getMessage());
        }
```

Основной файл сервера:



```java
package ru.oleg;


import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import ru.oleg.commands.*;
import ru.oleg.exceptions.ExitException;
import ru.oleg.managers.CollectionManager;
import ru.oleg.managers.CommandManager;
import ru.oleg.managers.FileManager;
import ru.oleg.utility.*;

import java.util.List;
import java.util.NoSuchElementException;
import java.util.Scanner;

public class App extends Thread {  Oleg346
    public static int PORT = 6086;  1 usage
    private static final Printable console = new PrintConsole();  1 usage
    private static final ConsoleOutput consoleOutput = new ConsoleOutput();  2 usages

    static final Logger rootLogger = LogManager.getRootLogger();  6 usages

    private static String getUserInput() {  1 usage  Oleg346
```



```
20      private static final ConsoleOutput consoleOutput = new ConsoleOutput();  2 usages
21
22      static final Logger rootLogger = LogManager.getRootLogger();  6 usages
23
24      private static String getUserInput() {  1 usage  Oleg346
25          Scanner scanner = new Scanner(System.in);
26          try {
27              return scanner.nextLine().trim();
28          } catch (NoSuchElementException e) {
29              return "";
30          }
31      }
32
33      public static void main(String[] args) {  Oleg346
34          CollectionManager collectionManager = new CollectionManager();
35          FileManager fileManager;
36
37          if (args.length > 0 && args[0] != null && !args[0].isEmpty()) {
38              // Первый аргумент командной строки используется в качестве пути к файлу
39              fileManager = new FileManager(consoleOutput, collectionManager, args[0]);
40          } else {
41              consoleOutput.printError( a: "Лее,друже, необходимо указать путь к файлу в качестве аргумента
42              return;
```

Вывод

В этой лабораторной работе я узнал об организации клиент-серверной архитектуры и реализовал её в своей программе. Познакомился с протоколами TCP/UDP, их работой, различием и реализацией, Stream API и моделью OSI. Узнал, что такое шаблоны проектирования, осуществил логирование различных этапов работы сервера.