

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

Отчёт

По лабораторной работе №6
по дисциплине «Программирование»

Вариант: 17128

Работу выполнил:

Поленов Кирилл Александрович

Группа Р3113

Работу принял:

Письмак Алексей Евгеньевич

г. Санкт-Петербург 2024

Оглавление

| | |
|--|---|
| Задание..... | 3 |
| Исходный код программы | 4 |
| UML диаграмма реализованной объектной модели | 4 |
| Выводы..... | 4 |

Задание

Лабораторная работа #6

Введите вариант: 17128

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 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 **потoki 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 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 **Log4J2**

Отчёт по работе должен содержать:

1. Текст задания.
2. Диаграмма классов разработанной программы (как клиентского, так и серверного приложения).
3. Исходный код программы.
4. Выводы по работе.

Исходный код программы

Репозиторий на GitHub:

<https://github.com/bilyardvmetro/ITMO-System-Application-Software/tree/main/%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%B5/2%20sem/Lab6/lab6>

UML диаграмма проекта

Репозиторий на GitHub:

<https://github.com/bilyardvmetro/ITMO-System-Application-Software/tree/main/%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%B5/2%20sem/Lab6/lab6>

Выводы

В ходе данной лабораторной работы я:

- Познакомился с сетевым программированием
- Освоил лямбда-выражения
- Использовал систему сборки Apache Maven для создания модулей и их конфигурации
- Использовал логгер Log4j2 и задал ему конфигурацию через XML файл
- Работал с каналами на сервере в неблокирующем режиме с помощью класса Selector
- Работал с байтовыми буферами