|  |
| --- |
|  |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **"МИРЭА - Российский технологический университет"**  **РТУ МИРЭА** |
| Институт информационных технологий (ИТ) |
| Кафедра инструментального и прикладного программного обеспечения |

|  |  |
| --- | --- |
| **ОТЧЕТ**  **ПО ПРАКТИЧЕСКОЙ РАБОТЕ №3** | |
| **по дисциплине** |  |
| **«Архитектура клиент-серверных приложений»**  **Тема: «Программирование JAVA сокетов»** | |
| Выполнил студент группы: ИКБО-20-21 | Костенко П.А. |
| Принял преподаватель кафедры ИиППО | Волков М.Ю. |

|  |  |  |
| --- | --- | --- |
| Практические работы выполнены | « » 2023 г. |  |
| «Зачтено» | « » 2023 г. |  |

Москва 2023

**Теоретическое введение**

Клиент-серверная архитектура — это самая известная структура приложений в Интернете. В этой архитектуре клиенты (например, персональные компьютеры, устройства IoT и т. Д.) вначале запрашивают ресурсы с сервера, а затем сервер отправляет соответствующие ответы на запросы клиентов. Чтобы это произошло, как на клиентах, так и на серверах должен быть реализован какой-то механизм, поддерживающий эту сетевую транзакцию. Этот механизм называется коммуникацией через сокеты.

Стоит отметить, что существует два типа сокетов для TCP и UDP. Поскольку большинство сетевых приложений используют TCP**.** Основное различие между ними заключается в том, что UDP не имеет соединения, то есть между клиентом и сервером нет сеанса, в то время как TCP ориентирован на соединение, то есть сначала должно быть установлено эксклюзивное соединение между клиентом и сервером для связи.

Сокет работает по типичной модели запрос / ответ, где в java-программе, называемой клиентом, вызывается другая программа, называемая сервером, работающая на другой JVM. Клиент должен отправить запрос, и сервер отвечает ответом. В этой модели обмен всегда инициируется клиентом; сервер не может отправлять какие-либо данные без предварительного запроса клиента. Стоит отметить, что данная модель хорошо работала во всемирной паутине, когда клиенты время от времени запрашивали документы, которые нечасто менялись, но ограничения этого подхода становятся все более актуальными, поскольку контент меняется быстро и пользователи ожидают более взаимодействия в сети. Например, WebSocket устраняет эти ограничения, предоставляя полнодуплексный канал связи между клиентом и сервером. Протокол WebSocket (JSR 356) будет рассмотрен в следующей практической работе. Но на практике, сокеты активно используются при реализации клент-серверного взаимодействия.

Сокет — это одна из конечных точек двустороннего канала связи между двумя программами, работающими в сети. Сокет привязан к номеру порта, чтобы уровень TCP мог идентифицировать приложение, данные, предназначенные для отправки.

Каждый сервер – это программа, которая работает в определенной системе и прослушивает определенный порт. Сокеты привязаны к номерам портов, и когда мы запускаем любой сервер, он просто слушает сокет и ждет клиентских запросов. Например, сервер tomcat, работающий на порту 8080, ожидает клиентских запросов и, получив любой клиентский запрос, отвечает на них.

Следовательно, для запуска любого сокета необходим запускаемый порт. Очевидно, что он размещен на машине, идентифицируемой именем хоста и уникальной адресованной IP.

В Java классы для поддержки программирования сокетов упакованы в пакет java.net. В соответствии с Java, [java.net](https://docs.oracle.com/javase/8/docs/api/java/net/package-summary.html) пакет поставляется с двумя классами Socket и ServerSocket для функций клиента и сервера соответственно.

В общем виде, для связи через сокет требуется серверный сокет, привязанный к порту хоста. На приведенной выше диаграмме показано, что сервер запускает серверный сокет с портом и прослушивает входящий запрос. После поступления запроса для клиента создается уникальный сокет, а потоки ввода и вывода используются для взаимодействия с каналом, созданным для клиента. На стороне клиента есть еще один сокет для инициирования соединения с сервером и отправки запроса.

**Постановка задачи**

Необходимо создать клиент-серверное приложение на языке JAVA с использованием socket, для широковещательного общения пользователей. Приложение может быть как консольным, так и оснащённым полноценным GUI. Клиентское приложение считывает данные из стандартного ввода и отсылает сообщение серверу (с помощью TCP/IP). Сервер, в свою очередь, накапливает сообщения и раз в 5 секунд осуществляет массовую рассылку всем клиентам. Если сообщений за указанный период не поступило, то рассылка не производится. Клиент, получивший сообщение, отображает на экране текст данного сообщения. Структуру и поведение данного клиент-серверного приложения, в том числе, **например**, в части регистрации конкретного клиента и формата широковещательного сообщения, студент определяет самостоятельно.

**Программный код**

На рисунках 1-2 представлена реализация серверной части.

****

Рисунок 1 – класс Server

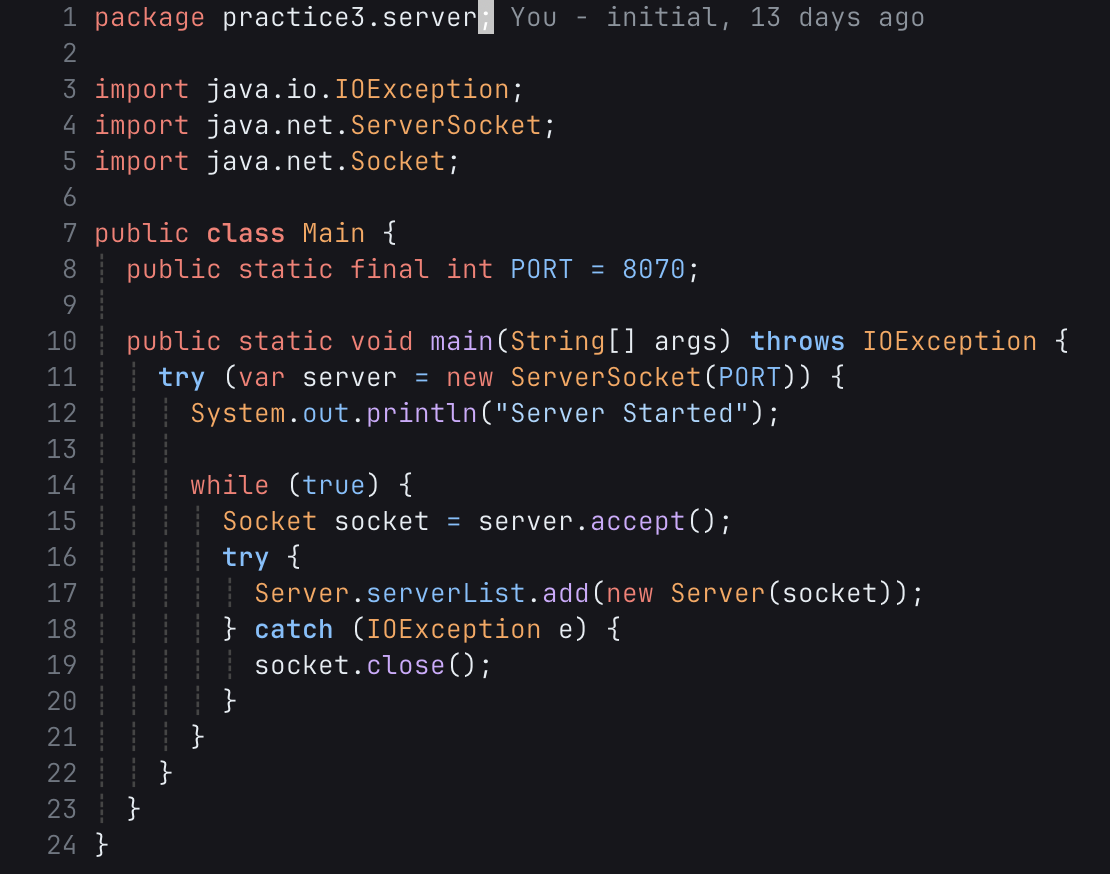
******

Рисунок 2 – Класс Main сервера

На рисунках 3–4 представлена реализация клиента.



Рисунок 3 – Класс Client

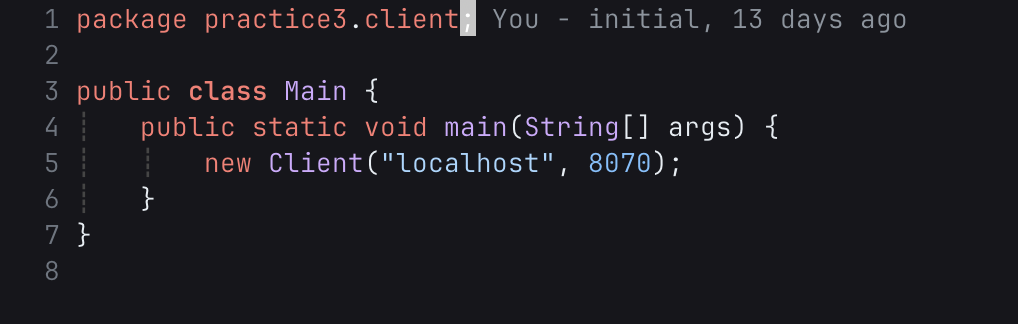


Рисунок 4 – Класс Main клиента

**Вывод программы**

На рисунках 5–7 представлена работа приложения.

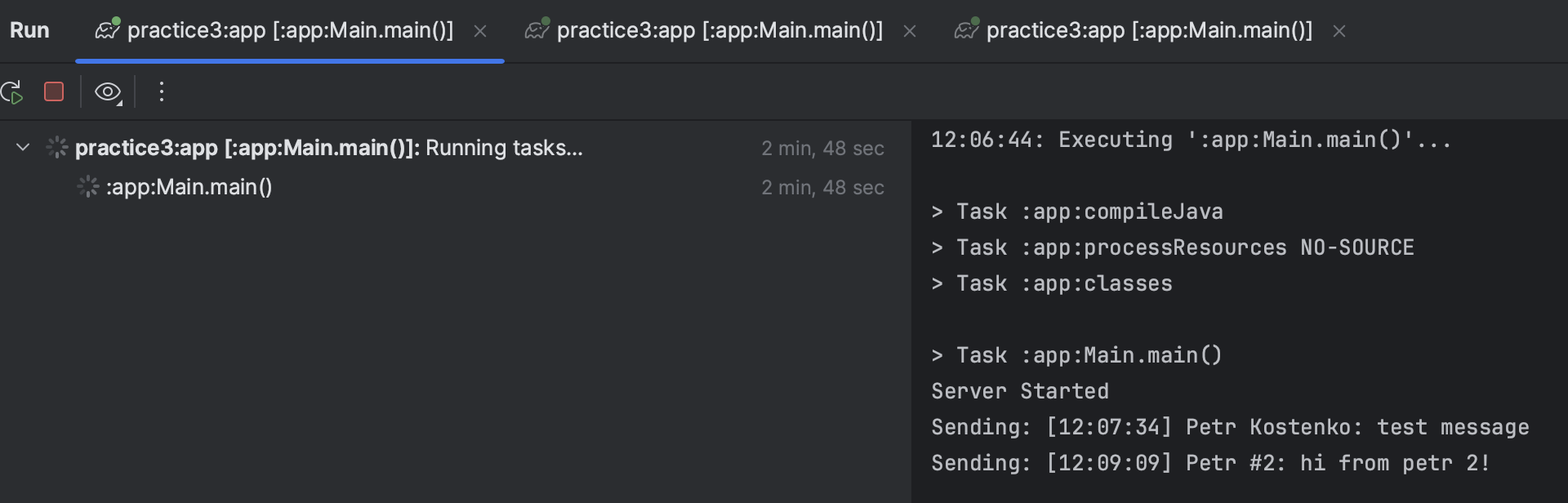


Рисунок 5 – Работа сервера

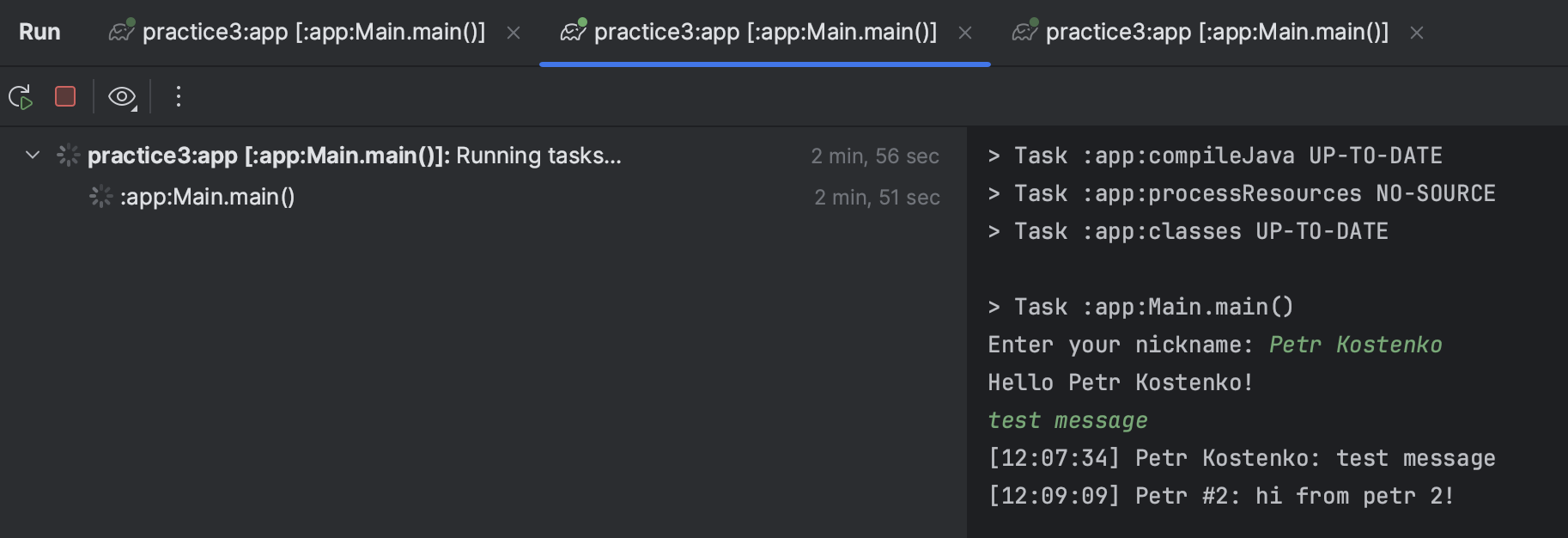


Рисунок 6 – Работа первого клиента

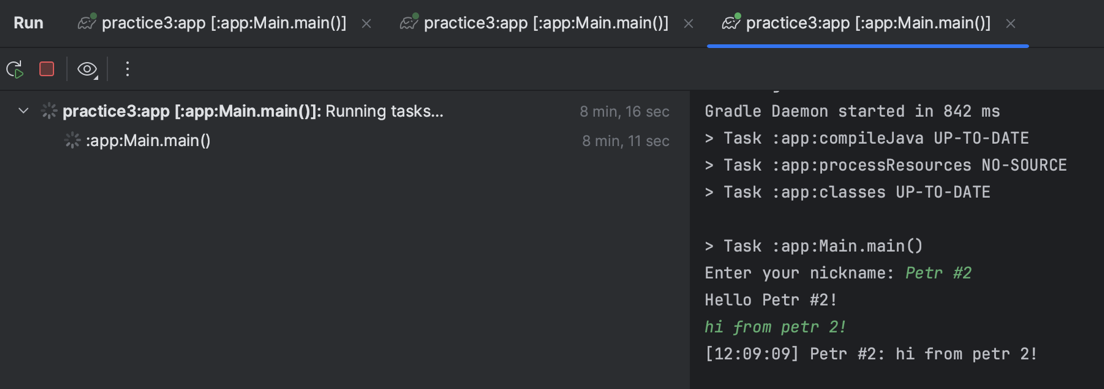


Рисунок 7 – Работа второго клиента

**Вывод**

В результате выполнения работы были освоены навыки работы с сокетами в Java. И реализовано приложение для широковещательного общения пользователей.

**Список литературы**

1. Документация Java 17 от Oracle [Электронный ресурс]: URL – <https://docs.oracle.com/en/java/javase/17/docs/api/index.html> [Дата обращения 7.10.2023];
2. Руководство по языку программирования Java [Электронный ресурс]: URL – <https://metanit.com/java/tutorial/> [Дата обращения 7.10.2023];
3. Socket (Java Platform SE 8) [Электронный ресурс]: URL - <https://docs.oracle.com/javase/8/docs/api/java/net/Socket.html> [Дата обращения 7.10.2023];
4. Классы Socket и ServerSocket [Электронный ресурс]: URL – <https://javarush.com/groups/posts/654-klassih-socket-i-serversocket-ili-allo-server-tih-menja-slihshishjh> [Дата обращения 7.10.2023];