|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА**  **Институт**информационных технологий  **Кафедры**Вычислительной техники |
|  |

**КУРСОВАЯ РАБОТА**

По дисциплине «Разработка клиент-серверных приложений»

**Тема курсовой работы** Клиент-сервеный чат на основе Netty\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Студент группы** ИКБО-14-18 Прушинский Иван Константинович \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(учебная группа, Фамилия Имя Отчество студента). (подпись студента)

**Руководитель курсовой работы** ст. преп. ВТ, Воронков С.О., \_\_\_\_\_\_\_\_\_\_\_\_\_

Должность, звание, ученая степень (подпись руководителя)

Работа представлена к защите «\_26\_» \_декабря\_ 2020 г.

Допущен к защите «\_26\_» \_ декабря\_ 2020 г.

**Задание**

Написать клиент-серверный чат, используя неблокирующую среду Netty. Чат должен поддерживать подключение нескольких пользователей, сообщения одного пользователя видны остальным, проверять наличие дублируемых имён пользователей, сообщать о подключении и отключении пользователей.

**Реализация**

**Сервер**

В основе сервера чата лежит обычный сервер Netty. В потоке обработки данных используются стандартные кодеры и декодеры строк (StringEncoder, StringDecoder), а также пользовательский класс MainHandler.

MainHandler представляет собой класс, наследующий дженерик с типом String SimpleChannelInboundHandler. В классе реализованы методы channelActive, channelRead0 и channelInactive, broadcastMessage, exeptionCaught. Также класс имеет поля channels (список всех пользователей), clientName (Имя пользователя), newClientIndex(счётчик пользователей).

channelActive запускается при первом подключении клиента. Пишет в чат о подключении нового пользователя.

В channelRead0 реализуется основная логика чата. Когда один из пользователей хочет сменить своё имя, он вводит команду /rename <новое имя>. Также сообщение, которое отправит пользователь отправляется остальным пользователям происходит в этом методе в вызове функции broadcastMessage.

В broadcastMessage реализовано отправление сообщения всем пользователям, кроме отправителя.

Если пользователь вышел из чата, то активируется функция channelInactive, которая пишет в чат о выходе пользователя из чата и удаляет его из channels.

Метод exceptionCaught отвечает за действия, если пользователь отвалился от чата. Если это произошло, то метод выведет в чат соответствующие сообщение.

**Клиент**

В основе клиента чата лежит стандартный сервер Netty. Клиент использует Java FX. Был разработан интерфейс с текстовым полем TextArea для вывода сообщений пользователей, и TextField для ввода сообщений пользователям, а также кнопка отправитиь. В потоке обработки данных используются стандартные кодеры и декодеры строк (StringEncoder, StringDecoder), а также пользовательский класс ClientHandler.

ClientHandler представляет собой класс, наследующий дженерик с типом String SimpleChannelInboundHandler. В классе реализованы методы channelRead0, ClientHander и exceptionCaught. А также поле класса Callback.

В конструкторе ClientHander получаем экземпляр класса Callback и записываем его.

В методе channelRead0, если Callback не равен NULL, то вызываем метод callback, который прописан в классе Controller, данный метод выводит сообщение в чат.

Метод exceptionCaught отвечает за выход пользователя из чата.

Класс Controller отвечает за взаимодействия объектов (кнопака, TextField, TextArea).

Класс Network отвечает за подключение к серверу.

Класс Main отвечает за создание окна и его параметров.

Сервер в качестве аргументов командной строки принимает на вход номер порта, а клиент – адрес хоста и номер порта.

**Результат работы**

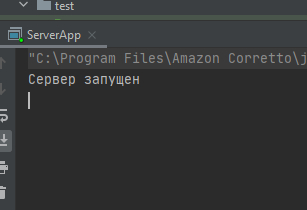


Рис.1. запуск сервера.

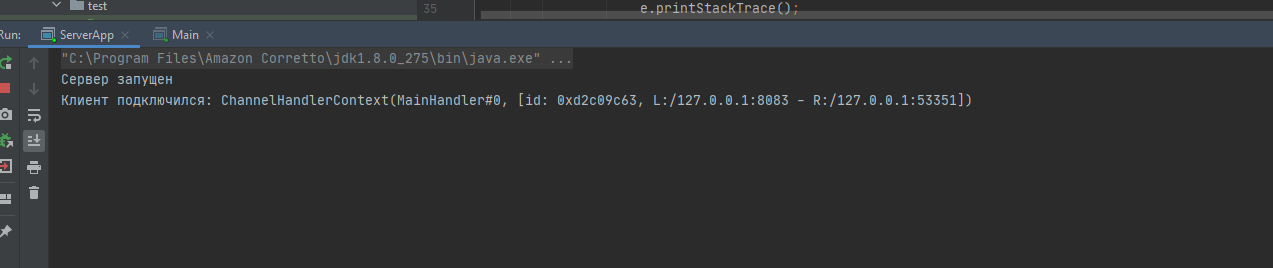


Рис.2. Подключение клиента к серверу.

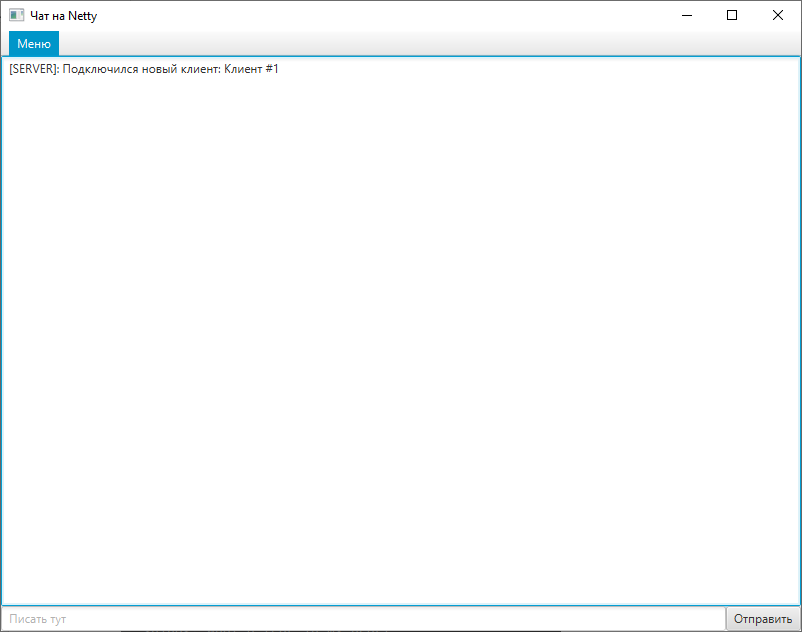


Рис.3. Запуск приложения клиента.

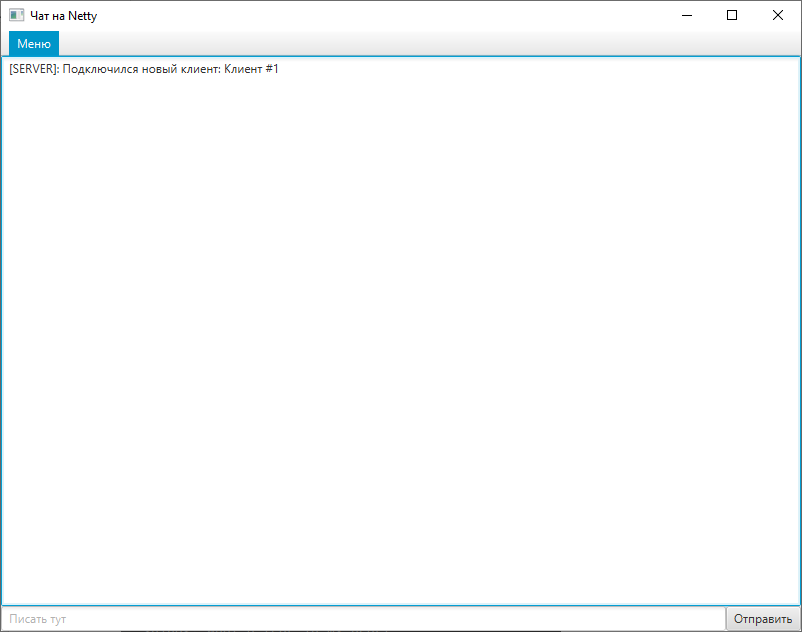


Рис.4. Замена имени пользователя.

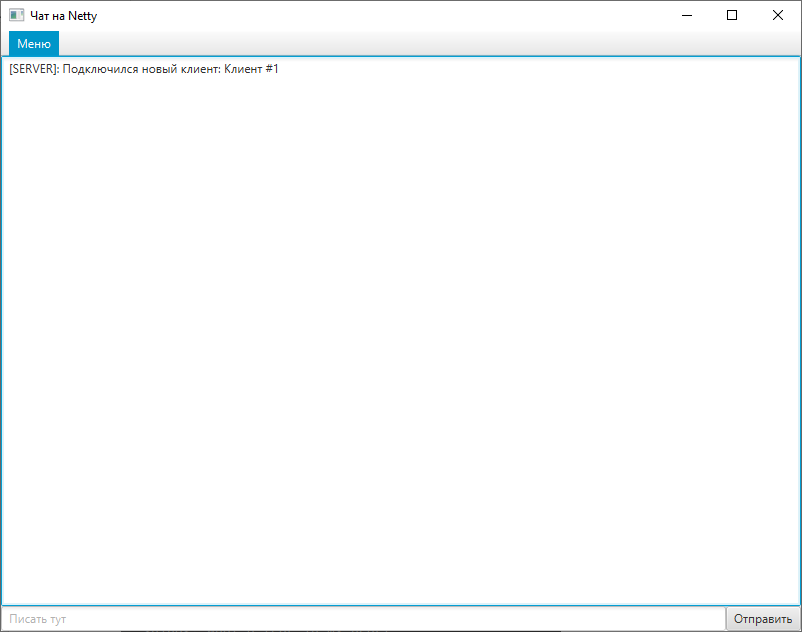


Рис.5. Подключение второго клиента.

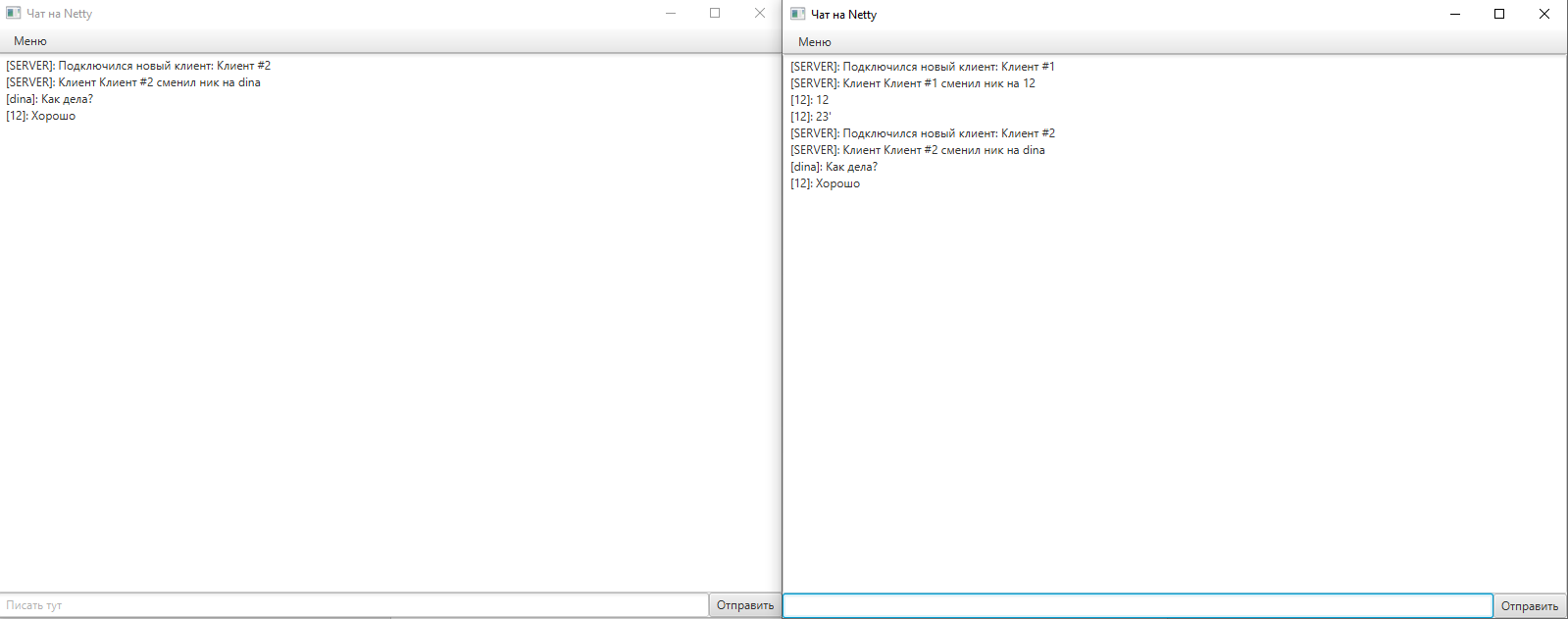


Рис.6. Общение двух клиентов.

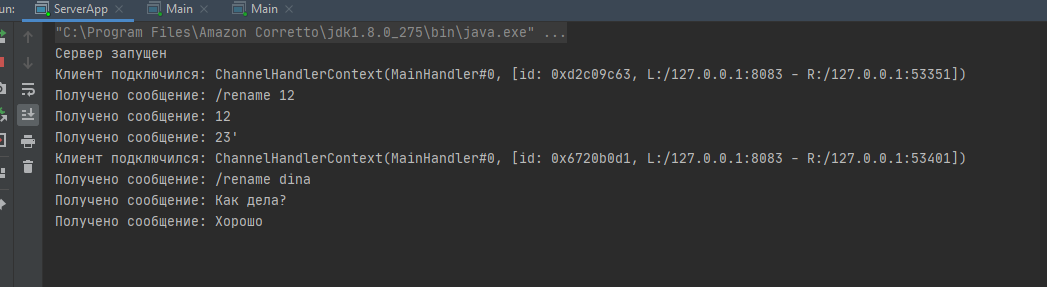


Рис.7. Дублирование на сервере чата.

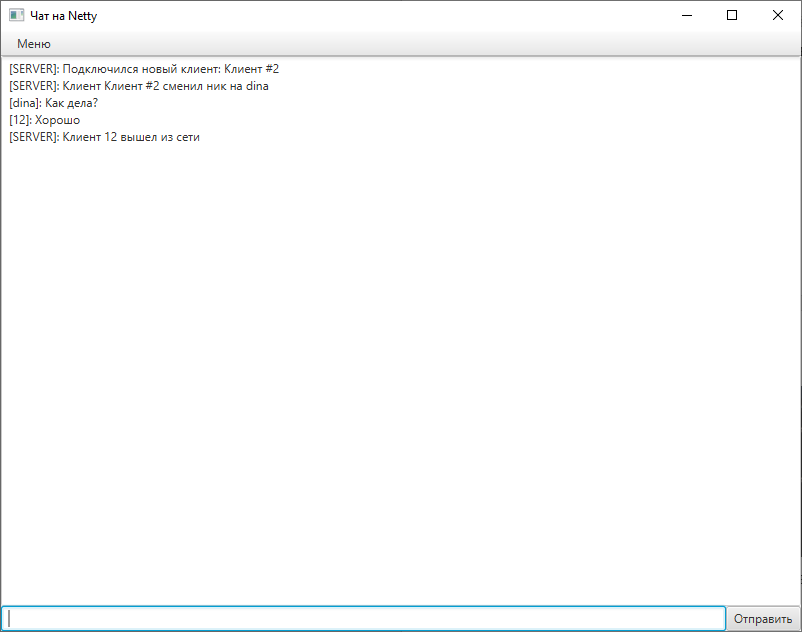


Рис.8. Выход из чата одно из пользователей.

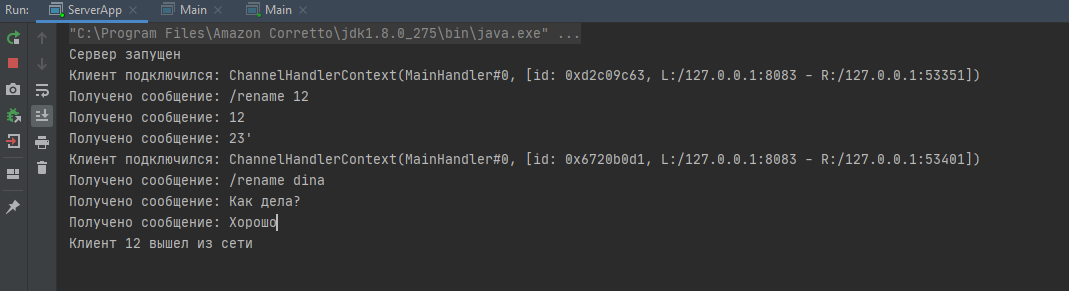


Рис.9. Отображение на сервере выхода из чата.