Министерство науки и высшего образования РФ

ФГАОУ ВО «Омский государственный технический университет»

Кафедра «Автоматизированные системы обработки информации и управления»

# ОТЧЁТ

По лабораторной работе №5

По дисциплине: Объектно-ориентированное программирование студента Попова Даниила Владимировича группы ПИН-212

# Пояснительная записка

Шифр работы От-2068998-43-ПИН-212-8 ЛР

Специальность 09.03.04

Старший преподаватель А.А. Кабанов

Студент Д.В. Попов

Омск 2022

**Цель работы:** в лабораторной работе №5 необходимо реализовать сетевое приложение, использующее протоколы стека TCP/IP.

**Задание:** Чат. Сервер рассылает всем пользователям информацию о пользователях вошедших в чат и покинувших его.

**Ход работы**

1. Листинг кода программы

import java.io.IOException;  
import java.io.PrintWriter;  
import java.net.ServerSocket;  
import java.net.Socket;  
import java.util.Set;  
import java.util.HashSet;  
import java.util.Scanner;  
import java.util.concurrent.Executors;  
  
  
public class server {  
 // Все имена клиентов, для проверки на уникальность.  
 private static Set <String> *names* = new HashSet<>();  
  
 // Набор всех средств печати для всех клиентов, используемых для широковещательной передачи.  
 private static Set <PrintWriter> *writers* = new HashSet<>();  
  
 public static void main(String[] args) throws Exception {  
 System.*out*.println("Сервер чата был успешно запущен.");  
  
 // Если передано более 500 потоков, они удерживаются в очереди, пока потоки не станут доступными.  
 var pool = Executors.*newFixedThreadPool*(500);  
 try (var listener = new ServerSocket(80)) {  
 while (true) {  
 pool.execute(new Handler(listener.accept()));  
 }  
 }  
 }  
  
  
 private static class Handler implements Runnable {  
 private String name;  
 private Socket socket;  
 private Scanner in;  
 private PrintWriter out;  
  
 // Создает поток-обработчик, удаляющий сокет.  
 public Handler(Socket socket) {  
 this.socket = socket;  
 }  
  
 public void run() {  
 try {  
 // переменная считывания  
 in = new Scanner(socket.getInputStream());  
  
 // переменная вывода  
 out = new PrintWriter(socket.getOutputStream(), true);  
  
 // Запрашивает имя, пока оно не будет уникальным  
 while (true) {  
 out.println("Введите ваше имя: ");  
 name = in.nextLine();  
 if (name == null) {  
 return;  
 }  
  
 // synchronized - в одно время может находится только 1 поток  
 synchronized (*names*) {  
 if (!name.isBlank() && !*names*.contains(name)) {  
 *names*.add(name);  
 break;  
 }  
 }  
 }  
  
 out.println("Добро пожаловать! " + name);  
 for (PrintWriter writer : *writers*) {  
 writer.println("СООБЩЕНИЕ " + name + " присоеденился к чату");  
 }  
 *writers*.add(out);  
  
 // Принимает сообщения от пользователя и транслирует его.  
 while (true) {  
 String input = in.nextLine();  
 if (input.toLowerCase().startsWith("/quit")) {  
 return;  
 }  
 for (PrintWriter writer : *writers*) {  
 writer.println("СООБЩЕНИЕ " + name + ": " + input);  
 }  
 }  
 } catch (Exception e) {  
 System.*out*.println(e);  
 } finally {  
 if (out != null) {  
 *writers*.remove(out);  
 }  
 if (name != null) {  
 System.*out*.println(name + " отключен");  
 *names*.remove(name);  
 for (PrintWriter writer : *writers*) {  
 writer.println("СООБЩЕНИЕ " + name + " покинул чат");  
 }  
 }  
 try {  
 socket.close();  
 } catch (IOException e) {  
 }  
 }  
 }  
 }

1. Тестирование кода

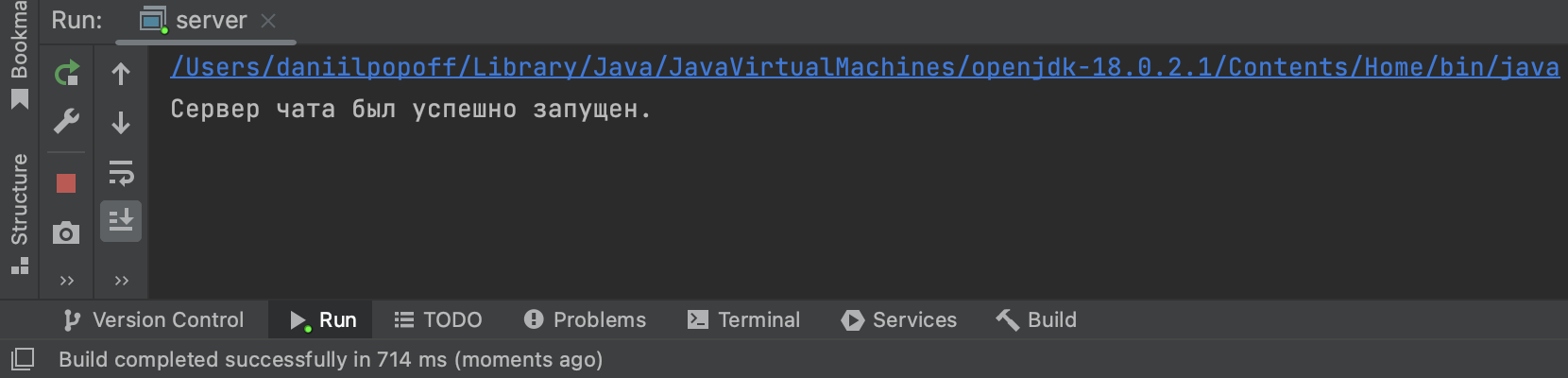


Рис. 1. – Сервер запущен.

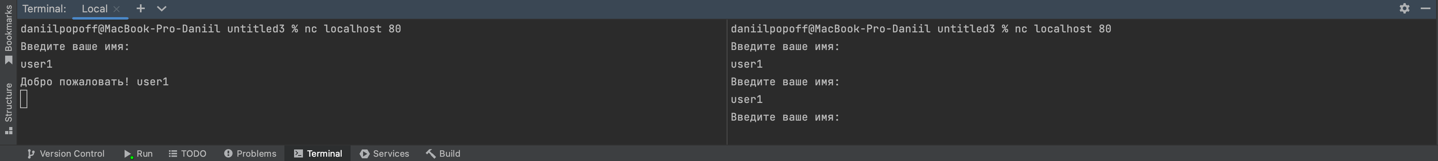


Рис. 2. – Попытка логина дубликата.

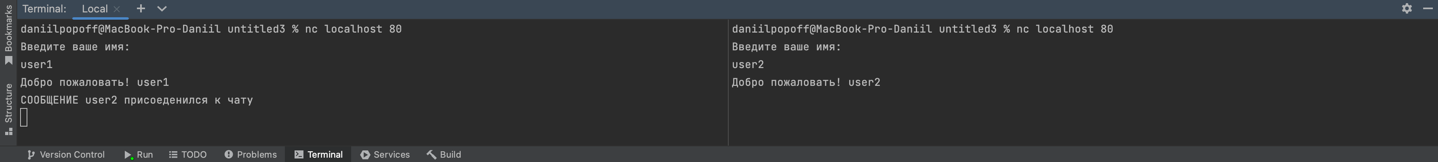


Рис. 3. – Логин (user1, user2).

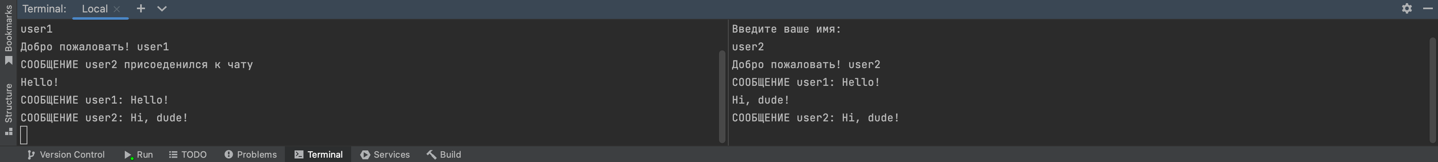


Рис. 4. – Имитация диалога юзеров.

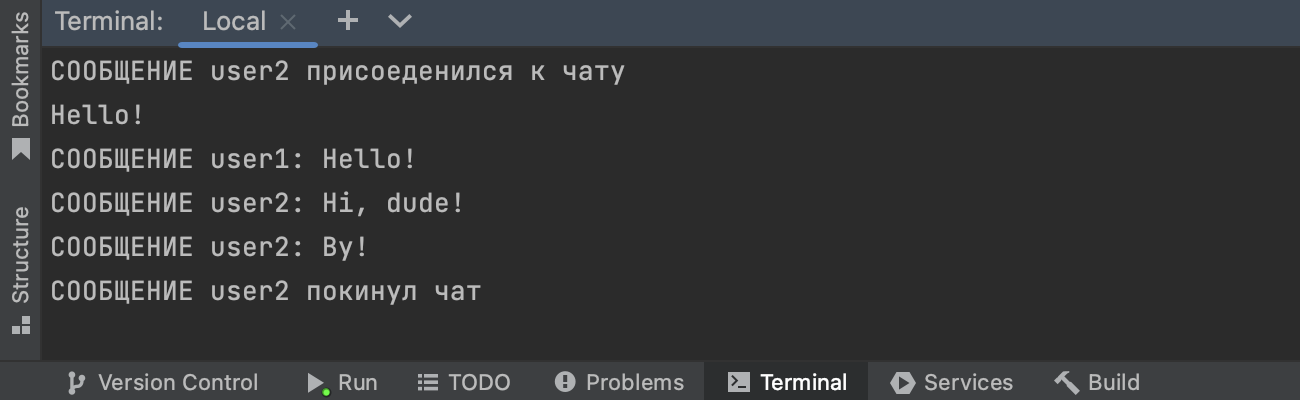


Рис. 5. – user2 покинул чат.

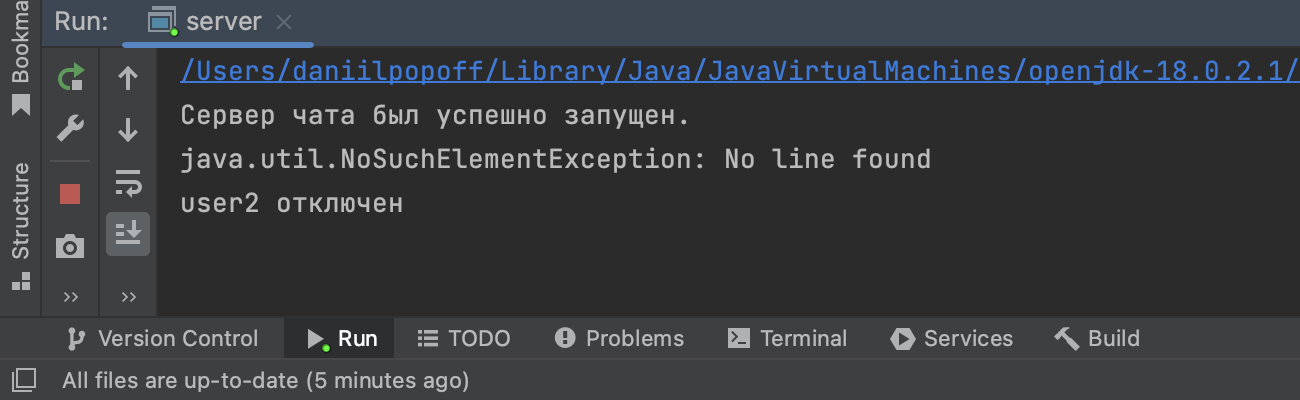


Рис. 6. – Вывод сервера (user2 покинул чат).

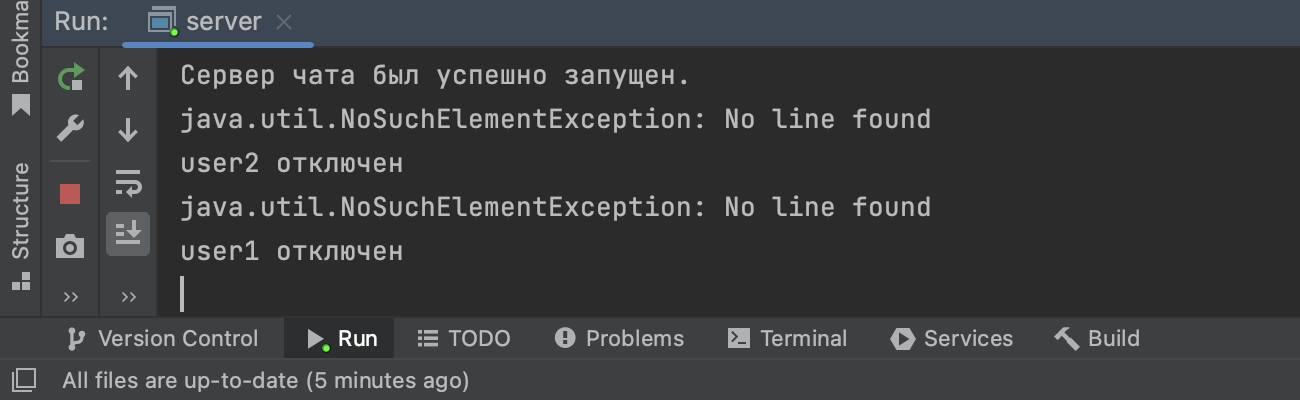


Рис. 7. – Вывод сервера (user2, user1 покинули чат).

**Вывод:** во время выполнения лабораторное работы было реализовано сетевое приложение, использующее протоколы стека TCP/IP, и выполнено задание, в результате которого сервер рассылает всем клиентам информацию о клиентах вошедших в чат и покинувших его.