import java.io.\*;

import java.net.\*;

public class ChatServer {

private static final int PORT = 8080;

public static void main(String[] args) {

System.out.println("Server is starting...");

try (ServerSocket serverSocket = new ServerSocket(PORT)) {

System.out.println("Waiting for clients...");

// Accept connections from two clients

Socket client1 = serverSocket.accept();

System.out.println("Client 1 connected.");

Socket client2 = serverSocket.accept();

System.out.println("Client 2 connected.");

// Create threads to handle communication

new Thread(new ClientHandler(client1, client2)).start();

new Thread(new ClientHandler(client2, client1)).start();

System.out.println("Chat is ready.");

} catch (IOException e) {

e.printStackTrace();

}

}

static class ClientHandler implements Runnable {

private Socket sender;

private Socket receiver;

public ClientHandler(Socket sender, Socket receiver) {

this.sender = sender;

this.receiver = receiver;

}

@Override

public void run() {

try (

BufferedReader senderInput = new BufferedReader(new InputStreamReader(sender.getInputStream()));

BufferedWriter receiverOutput = new BufferedWriter(new OutputStreamWriter(receiver.getOutputStream()))

) {

String message;

while ((message = senderInput.readLine()) != null) {

System.out.println("Message received: " + message);

receiverOutput.write("Message from peer: " + message + "\n");

receiverOutput.flush();

}

} catch (IOException e) {

System.out.println("A client disconnected.");

}

}

}

}  
  
  
  
  
  
  
  
  
  
  
  
import java.io.\*;

import java.net.\*;

public class ChatClient {

private static final String SERVER\_ADDRESS = "localhost";

private static final int SERVER\_PORT = 8080;

public static void main(String[] args) {

try (

Socket socket = new Socket(SERVER\_ADDRESS, SERVER\_PORT);

BufferedReader serverInput = new BufferedReader(new InputStreamReader(socket.getInputStream()));

BufferedWriter serverOutput = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()));

BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in))

) {

System.out.println("Connected to the chat server.");

// Start a thread to listen for messages from the server

Thread listener = new Thread(() -> {

try {

String message;

while ((message = serverInput.readLine()) != null) {

System.out.println(message);

}

} catch (IOException e) {

System.out.println("Disconnected from server.");

}

});

listener.start();

// Read user input and send it to the server

String userMessage;

while ((userMessage = userInput.readLine()) != null) {

serverOutput.write(userMessage + "\n");

serverOutput.flush();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}