// Client Side

import java.io.\*;

import java.net.\*;

public class TCPClient {

public void run() {

try {

int serverPort = 4020;

InetAddress host = InetAddress.getByName("localhost");

System.out.println("Connecting to server on port " + serverPort);

Socket socket = new Socket(host,serverPort);

//Socket socket = new Socket("127.0.0.1", serverPort);

System.out.println("Just connected to " + socket.getRemoteSocketAddress());

PrintWriter toServer =

new PrintWriter(socket.getOutputStream(),true);

BufferedReader fromServer =

new BufferedReader(

new InputStreamReader(socket.getInputStream()));

toServer.println("Hello from " + socket.getLocalSocketAddress());

String line = fromServer.readLine();

System.out.println("Client received: " + line + " from Server");

toServer.close();

fromServer.close();

socket.close();

}

catch(UnknownHostException ex) {

ex.printStackTrace();

}

catch(IOException e){

e.printStackTrace();

}

}

public static void main(String[] args) {

TCPClient client = new TCPClient();

client.run();

}

}

// Server Side

import java.net.\*;

import java.io.\*;

public class TCPServer {

public void run() {

try {

int serverPort = 4020;

ServerSocket serverSocket = new ServerSocket(serverPort);

serverSocket.setSoTimeout(10000);

while(true) {

System.out.println("Waiting for client on port " + serverSocket.getLocalPort() + "...");

Socket server = serverSocket.accept();

System.out.println("Just connected to " + server.getRemoteSocketAddress());

PrintWriter toClient =

new PrintWriter(server.getOutputStream(),true);

BufferedReader fromClient =

new BufferedReader(

new InputStreamReader(server.getInputStream()));

String line = fromClient.readLine();

System.out.println("Server received: " + line);

toClient.println("Thank you for connecting to " + server.getLocalSocketAddress() + "\nGoodbye!");

}

}

catch(UnknownHostException ex) {

ex.printStackTrace();

}

catch(IOException e){

e.printStackTrace();

}

}

public static void main(String[] args) {

TCPServer srv = new TCPServer();

srv.run();

}

}

**// DatagramServer**

import java.net.\*;

public class DatagramServer {

public static void main(String[] args) {

try {

// Create a datagram socket

DatagramSocket serverSocket = new DatagramSocket(9876);

byte[] receiveData = new byte[1024];

byte[] sendData = new byte[1024];

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

while (true) {

// Receive packet from client

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

serverSocket.receive(receivePacket);

String receivedMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());

System.out.println("Received from client: " + receivedMessage);

// Get client address and port

InetAddress clientAddress = receivePacket.getAddress();

int clientPort = receivePacket.getPort();

// Prepare data to send back to client

String responseMessage = "Message received by server";

sendData = responseMessage.getBytes();

// Create packet to send to client

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress, clientPort);

// Send packet

serverSocket.send(sendPacket);

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**//DatagramClient**

import java.net.\*;

public class DatagramClient {

public static void main(String[] args) {

try {

// Create a datagram socket

DatagramSocket clientSocket = new DatagramSocket();

byte[] sendData;

byte[] receiveData = new byte[1024];

InetAddress serverAddress = InetAddress.getByName("localhost");

int serverPort = 9876;

// Message to be sent to the server

String message = "Hello, server!";

sendData = message.getBytes();

// Create packet to send to server

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort);

// Send packet to server

clientSocket.send(sendPacket);

System.out.println("Message sent to server: " + message);

// Receive response from server

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

clientSocket.receive(receivePacket);

String receivedMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());

System.out.println("Received from server: " + receivedMessage);

// Close socket

clientSocket.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}