**UDP Server**

```
import java.io.*;
import java.net.*;
public class UDPServer {
public static void main(String[] args) {
DatagramSocket serverSocket = null;
try {
// Create a UDP socket
serverSocket = new DatagramSocket(9876);
byte[] receiveData = new byte[1024];
System.out.println("Server is waiting for data...");27
while (true) {
DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
serverSocket.receive(receivePacket);
String clientMessage = new String(receivePacket.getData(), 0,
receivePacket.getLength());
InetAddress clientAddress = receivePacket.getAddress();
int clientPort = receivePacket.getPort();
System.out.println("Received from client at " + clientAddress + ":" + clientPort + ":
" + clientMessage);
// Process the received data (you can add your logic here)
// Send a reply back to the client
String serverReply = "Hello from the server!";
byte[] sendData = serverReply.getBytes();
DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
clientAddress, clientPort);
serverSocket.send(sendPacket);
}
} catch (IOException e) {
e.printStackTrace();
} finally {
if (serverSocket != null && !serverSocket.isClosed()) {
serverSocket.close();
}
}
}
}
```

**UDP Client**

```
import java.io.*;
import java.net.*;
public class UDPClient {
public static void main(String[] args)
{
DatagramSocket clientSocket = null;
try {
// Create a UDP socket
clientSocket = new DatagramSocket();
InetAddress serverAddress = InetAddress.getByName("127.0.0.1");
int serverPort = 9876;
BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));
while (true) {
System.out.print("Enter a message to send to the server (or 'exit' to quit): ");
String message = userInput.readLine();
if (message.equals("exit")) {
break;28
}
byte[] sendData = message.getBytes();
DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
serverAddress, serverPort);
clientSocket.send(sendPacket);
byte[] receiveData = new byte[1024];
DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
clientSocket.receive(receivePacket);
String serverReply = new String(receivePacket.getData(), 0,
receivePacket.getLength());
System.out.println("Received from server: " + serverReply);
}
} catch (IOException e) {
e.printStackTrace();
} finally {
if (clientSocket != null && !clientSocket.isClosed()) {
clientSocket.close();
}
}
}
```