

(01)

UDP Server

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class Server {
    public static void main(String[] args) {
        try {
            // Create UDP socket on port 8088
            DatagramSocket socket = new DatagramSocket(8088);
            System.out.println("Server started. Waiting for client...");

            byte[] receiveBuffer = new byte[1024];

            // Receive packet from client
            DatagramPacket receivePacket = new DatagramPacket(receiveBuffer, receiveBuffer.length);
            socket.receive(receivePacket);

            String clientMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());
            System.out.println("Client says: " + clientMessage);

            // Send response to client
            String response = "Hello, Client!";
            byte[] sendBuffer = response.getBytes();

            InetAddress clientAddress = receivePacket.getAddress();
            int clientPort = receivePacket.getPort();

            DatagramPacket sendPacket = new DatagramPacket(sendBuffer, sendBuffer.length,
                clientAddress, clientPort);
        }
    }
}
```

```
        socket.send(sendPacket);

        System.out.println("Response sent to client.");

        socket.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

UDP Client

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class Client {
    public static void main(String[] args) {
        try {
            DatagramSocket socket = new DatagramSocket();

            String message = "Hello, Server!";
            byte[] sendBuffer = message.getBytes();

            InetAddress serverAddress = InetAddress.getByName("localhost");
            int serverPort = 8088;

            // Send message to server
            DatagramPacket sendPacket = new DatagramPacket(sendBuffer, sendBuffer.length,
                serverAddress, serverPort);
            socket.send(sendPacket);
            System.out.println("Message sent to server: " + message);
        }
    }
}
```

```
// Receive response  
  
byte[] receiveBuffer = new byte[1024];  
  
DatagramPacket receivePacket = new DatagramPacket(receiveBuffer, receiveBuffer.length);  
socket.receive(receivePacket);  
  
String serverResponse = new String(receivePacket.getData(), 0, receivePacket.getLength());  
System.out.println("Server says: " + serverResponse);  
  
socket.close();  
}  
catch (Exception e) {  
    e.printStackTrace();  
}  
}  
}
```

(03)

```
import java.net.DatagramPacket;  
import java.net.DatagramSocket;  
import java.net.InetAddress;  
import java.net.SocketTimeoutException;
```

```
public class ClientWithRetry {  
    public static void main(String[] args) {  
        try {  
            DatagramSocket socket = new DatagramSocket();  
            socket.setSoTimeout(3000); // 3 seconds timeout  
  
            String message = "Hello, Server!";  
            byte[] sendBuffer = message.getBytes();  
  
            InetAddress serverAddress = InetAddress.getByName("localhost");
```

```
int serverPort = 8088;

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

boolean receivedResponse = false;
int retries = 0;

while (!receivedResponse && retries < 3) {
    try {
        // Send message
        socket.send(sendPacket);
        System.out.println("Message sent to server (Attempt " + (retries + 1) + "): " + message);

        // Wait for response
        byte[] receiveBuffer = new byte[1024];
        DatagramPacket receivePacket = new DatagramPacket(receiveBuffer,
receiveBuffer.length);
        socket.receive(receivePacket);

        String serverResponse = new String(receivePacket.getData(), 0,
receivePacket.getLength());
        System.out.println("Server says: " + serverResponse);

        receivedResponse = true; // success
    } catch (SocketTimeoutException e) {
        retries++;
        System.out.println("Timeout! Retrying... (" + retries + ")");
    }
}

if (!receivedResponse) {
```

```
        System.out.println("No response from server after 3 attempts. Exiting.");
    }

    socket.close();
} catch (Exception e) {
    e.printStackTrace();
}
}

}
```