J**AVA NETWORKING PROGRAMS**

TCP CLIENT SERVER PROGRAM

Server.java

// A Java program for a Server

import java.net.\*;

import java.io.\*;

public class Server

{

    //initialize socket and input stream

    private Socket       socket = null;

    private ServerSocket server = null;

    private DataInputStream in   = null;

    private DataOutputStream out = null;

    // constructor with port

    public Server(int port)

    {

        // starts server and waits for a connection

        try

        {

            server = new ServerSocket(port);

            System.out.println("Server started");

            System.out.println("Waiting for a client ...");

            socket = server.accept();

            System.out.println("Client accepted");

            // takes input from the client socket

            in = new DataInputStream(

                new BufferedInputStream(socket.getInputStream()));

            out = new DataOutputStream(

                socket.getOutputStream());

            String line = "";

            // reads message from client until "Over" is sent

            while (!line.equals("Over"))

            {

                try

                {

                    line = in.readUTF();

                    System.out.println(line);

                    String rString= new StringBuffer(line).reverse().toString();

                    out.writeUTF(rString);

                }

                catch(IOException i)

                {

                    System.out.println(i);

                }

            }

            System.out.println("Closing connection");

            // close connection

            socket.close();

            in.close();

        }

        catch(IOException i)

        {

            System.out.println(i);

        }

    }

    public static void main(String args[])

    {

        Server server = new Server(5000);

    }

}

Client.java

// A Java program for a Client

import java.io.\*;

import java.net.\*;

public class Client {

    // initialize socket and input output streams

    private Socket socket = null;

    private DataInputStream input = null;

    private DataOutputStream out = null;

    private DataInputStream in   = null;

    // constructor to put ip address and port

    public Client(String address, int port)

    {

        // establish a connection

        try {

            socket = new Socket(address, port);

            System.out.println("Connected");

            // takes input from terminal

            input = new DataInputStream(System.in);

            in = new DataInputStream(

                new BufferedInputStream(socket.getInputStream()));

            // sends output to the socket

            out = new DataOutputStream(

                socket.getOutputStream());

        }

        catch (UnknownHostException u) {

            System.out.println(u);

            return;

        }

        catch (IOException i) {

            System.out.println(i);

            return;

        }

        // string to read message from input

        String line = "";

        String ln="";

        // keep reading until "Over" is input

        while (!line.equals("Over")) {

            try {

                line = input.readLine();

                out.writeUTF(line);

                ln=in.readUTF();

                System.out.println("REVERSED"+ln);

            }

            catch (IOException i) {

                System.out.println(i);

            }

        }

        // close the connection

        try {

            input.close();

            out.close();

            socket.close();

        }

        catch (IOException i) {

            System.out.println(i);

        }

    }

    public static void main(String args[])

    {

        Client client = new Client("127.0.0.1", 5000);

    }

}

UDP Client Server Program

// Demonstrate datagrams.

import java.net.\*;

class WriteServer {

  public static int serverPort = 998;

  public static int clientPort = 999;

  public static int buffer\_size = 1024;

  public static DatagramSocket ds;

  public static byte buffer[] = new byte[buffer\_size];

  public static void TheServer() throws Exception {

    int pos=0;

    while (true) {

      int c = System.in.read();

      switch (c) {

              case '\r':

                break;

              case '\n':

                ds.send(new DatagramPacket(buffer,pos,InetAddress.getLocalHost(),clientPort));

                pos=0;

                break;

              default:

                buffer[pos++] = (byte) c;

            }

          }

        }

        public static void TheClient() throws Exception {

          while(true) {

            DatagramPacket p = new DatagramPacket(buffer, buffer.length);

            ds.receive(p);

            System.out.println(new String(p.getData(), 0, p.getLength()));

          }

        }

        public static void main(String args[]) throws Exception {

          if(args.length == 1) {

            ds = new DatagramSocket(serverPort);

            TheServer();

          } else {

            ds = new DatagramSocket(clientPort);

            TheClient();

          }

        }

}