**1)TCPEchoClient**

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class TCPEchoClient

{

private static InetAddress host;

private static final int PORT = 1234;

public static void main(String[] args)

{

try

{

host = InetAddress.getLocalHost();

}

catch(UnknownHostException uhEx)

{

System.out.println("Host ID not found!");

System.exit(1);

}

accessServer();

}

private static void accessServer()

{

Socket link = null; //Step 1.

try

{

link = new Socket(host,PORT); //Step 1.

Scanner input =new Scanner(link.getInputStream()); //Step 2.

PrintWriter output =new PrintWriter(link.getOutputStream(),true); //Step 2.

//Set up stream for keyboard entry...

Scanner userEntry = new Scanner(System.in);

String message, response;

do

{

System.out.print("Enter message: ");

message = userEntry.nextLine();

output.println(message); //Step 3.

response = input.nextLine(); //Step 3.

System.out.println("\nSERVER> "+response);

}while (!message.equals("\*\*\*CLOSE\*\*\*"));

}

catch(IOException ioEx)

{

ioEx.printStackTrace();

}

finally

{

try

{

System.out.println("\n\* Closing connection... \*");

link.close(); //Step 4.

}

catch(IOException ioEx)

{

System.out.println("Unable to disconnect!");

System.exit(1);

}

}

}

}

**2)Server that echoes back TCP client's messages.**

//At end of dialogue, sends message indicating number of

//messages received. Uses TCP.

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class TCPEchoServer

{

private static ServerSocket servSock;

private static final int PORT = 1234;

public static void main(String[] args)

{

System.out.println("Opening port...\n");

try

{

servSock = new ServerSocket(PORT); //Step 1.

}

catch(IOException ioEx)

{

System.out.println("Unable to attach to port!");

System.exit(1);

}

do

{

handleClient();

}while (true);

}

private static void handleClient()

{

Socket link = null; //Step 2.

try

{

link = servSock.accept(); //Step 2.

Scanner input =new Scanner(link.getInputStream());//Step 3.

PrintWriter output =new PrintWriter(link.getOutputStream(),true); //Step 3.

int numMessages = 0;

String message = input.nextLine(); //Step 4.

while (!message.equals("\*\*\*CLOSE\*\*\*"))

{

System.out.println("Message received.");

numMessages++;

output.println("Message " + numMessages+ ": " + message); //Step 4.

message = input.nextLine();

}

output.println(numMessages+ " messages received.");//Step 4.

}

catch(IOException ioEx)

{

ioEx.printStackTrace();

}

finally

{

try

{

System.out.println("\n\* Closing connection... \*");

link.close(); //Step 5.

}

catch(IOException ioEx)

{

System.out.println("Unable to disconnect!");

System.exit(1);

}

}

}

}

**3) UDPEchoClient**

**import** java.io.\*;

**import** java.net.\*;

**import** java.util.\*;

**public** **class** UDPEchoClient

{

**private** **static** InetAddress *host*;

**private** **static** **final** **int** ***PORT*** = 1234;

**private** **static** DatagramSocket *datagramSocket*;

**private** **static** DatagramPacket *inPacket*, *outPacket*;

**private** **static** **byte**[] *buffer*;

**public** **static** **void** main(String[] args)

{

**try**

{

*host* = InetAddress.*getLocalHost*();

}

**catch**(UnknownHostException uhEx)

{

System.***out***.println("Host ID not found!");

System.*exit*(1);

}

*accessServer*();

}

**private** **static** **void** accessServer()

{

**try**

{

//Step 1...

*datagramSocket* = **new** DatagramSocket();

//Set up stream for keyboard entry...

Scanner userEntry = **new** Scanner(System.***in***);

String message="", response="";

**do**

{

System.***out***.print("Enter message: ");

message = userEntry.nextLine();

**if** (!message.equals("\*\*\*CLOSE\*\*\*"))

{

*outPacket* = **new** DatagramPacket(message.getBytes(),message.length(),*host*,***PORT***); //Step 2.

//Step 3...

*datagramSocket*.send(*outPacket*);

*buffer* = **new** **byte**[256]; //Step 4.

*inPacket* =**new** DatagramPacket(*buffer*, *buffer*.length);//Step 5.

//Step 6...

*datagramSocket*.receive(*inPacket*);

response =**new** String(*inPacket*.getData(),0, *inPacket*.getLength()); //Step 7.

System.***out***.println("\nSERVER> "+response);

}

}**while** (!message.equals("\*\*\*CLOSE\*\*\*"));

}

**catch**(IOException ioEx)

{

ioEx.printStackTrace();

}

**finally**

{

System.***out***.println("\n\* Closing connection... \*");

*datagramSocket*.close(); //Step 8.

}

}

}

**4)Server that echoes back UDPclient's messages.**

//At end of dialogue, sends message indicating number of

//messages received. Uses datagrams.

import java.io.\*;

import java.net.\*;

public class UDPEchoServer

{

private static final int PORT = 1234;

private static DatagramSocket datagramSocket;

private static DatagramPacket inPacket, outPacket;

private static byte[] buffer;

public static void main(String[] args)

{

System.out.println("Opening port...\n");

try

{

datagramSocket =new DatagramSocket(PORT); //Step 1.

}

catch(SocketException sockEx)

{

System.out.println("Unable to attach to port!");

System.exit(1);

}

handleClient();

}

private static void handleClient()

{

try

{

String messageIn,messageOut;

int numMessages = 0;

do

{

buffer = new byte[256]; //Step 2.

inPacket =new DatagramPacket(buffer, buffer.length); //Step 3.

datagramSocket.receive(inPacket);//Step 4.

InetAddress clientAddress =inPacket.getAddress(); //Step 5.

int clientPort =inPacket.getPort(); //Step 5.

messageIn =new String(inPacket.getData(),0,inPacket.getLength()); //Step 6.

System.out.println("Message received.");

numMessages++;

messageOut = "Message " + numMessages+ ": " + messageIn;

outPacket =new DatagramPacket(messageOut.getBytes(),messageOut.length(),clientAddress,clientPort); //Step 7.

datagramSocket.send(outPacket); //Step 8.

}while (true);

}

catch(IOException ioEx)

{

ioEx.printStackTrace();

}

finally //If exception thrown, close connection.

{

System.out.println("\n\* Closing connection... \*");

datagramSocket.close(); //Step 9.

}

}

}