**UDP**

package aula2distribuidos;

import java.net.\*;

import java.io.\*;

import java.util.Scanner;

public class UDPClient{

public static void main(String args[]){

// args give message contents and destination hostname

DatagramSocket aSocket = null;

try {

aSocket = new DatagramSocket();

//byte [] m = args[0].getBytes();

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

int serverPort = 6789;

while(true){

Scanner aux=new Scanner(System.in);

String a = aux.nextLine();

DatagramPacket request =

new DatagramPacket(a.getBytes(), a.length(), aHost, serverPort);

aSocket.send(request);

byte[] buffer = new byte[1000];

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

aSocket.setSoTimeout(5000);

aSocket.receive(reply);

System.out.println("Reply: " + new String(reply.getData()));

}

}catch (SocketException e){System.out.println("Socket: " + e.getMessage());

}catch (IOException e){System.out.println("IO: " + e.getMessage());

}finally {if(aSocket != null) aSocket.close();}

}

}

package aula2distribuidos;

import java.net.\*;

import java.io.\*;

public class UDPServer{

public static void main(String args[]){

DatagramSocket aSocket = null;

try{

aSocket = new DatagramSocket(6789);

// create socket at agreed port

byte[] buffer = new byte[1000];

while(true){

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

aSocket.receive(request);

DatagramPacket reply = new DatagramPacket(request.getData(), request.getLength(),

request.getAddress(), request.getPort());

aSocket.send(reply);

System.out.println("Reply: " + new String(reply.getData()));

}

}catch (SocketException e){System.out.println("Socket: " + e.getMessage());

}catch (IOException e) {System.out.println("IO: " + e.getMessage());

}finally {if(aSocket != null) aSocket.close();}

}

}

**TCP**

import java.net.\*;

import java.io.\*;

import java.util.Scanner;

public class TCPClient {

public static void main (String args[]) {

// arguments supply message and hostname

Socket s = null;

try{

int serverPort = 7896;

s = new Socket("localhost", serverPort);

while(true){

Scanner aux = new Scanner(System.in);

String a = aux.nextLine();

DataInputStream in = new DataInputStream( s.getInputStream());

DataOutputStream out =new DataOutputStream( s.getOutputStream());

out.writeUTF(a); // UTF is a string encoding see Sn. 4.4

String data = in.readUTF(); // read a line of data from the stream

System.out.println("Received: "+ data) ;

}

}catch (UnknownHostException e){System.out.println("Socket:"+e.getMessage());

}catch (EOFException e){System.out.println("EOF:"+e.getMessage());

}catch (IOException e){System.out.println("readline:"+e.getMessage());

}finally {if(s!=null) try {s.close();}catch (IOException e){System.out.println("close:"+e.getMessage());}}

}

}

import java.net.\*;

import java.io.\*;

public class TCPServer {

public static void main (String args[]) {

try{

int serverPort = 7896; // the server port

ServerSocket listenSocket = new ServerSocket(serverPort);

while(true) {

Socket clientSocket = listenSocket.accept();

Connection c = new Connection(clientSocket);

}

} catch(IOException e) {System.out.println("Listen socket:"+e.getMessage());}

}

}

class Connection extends Thread {

DataInputStream in;

DataOutputStream out;

Socket clientSocket;

public Connection (Socket aClientSocket) {

try {

clientSocket = aClientSocket;

in = new DataInputStream( clientSocket.getInputStream());

out =new DataOutputStream( clientSocket.getOutputStream());

this.start();

} catch(IOException e) {System.out.println("Connection:"+e.getMessage());}

}

public void run(){

try { // an echo server

String data = in.readUTF(); // read a line of data from the stream

out.writeUTF(data);

System.out.println(data);

}catch (EOFException e){System.out.println("EOF:"+e.getMessage());

} catch(IOException e) {System.out.println("readline:"+e.getMessage());

} finally{ try {clientSocket.close();}catch (IOException e){/\*close failed\*/}}

}

}

**MULTICAST**

package multicastclient;

import java.io.IOException;

import java.net.\* ;

import java.util.\* ;

public class MulticastClient {

public static void main(String argv[]) throws Exception

{

final Processo processo = new Processo();

Scanner scanner = new Scanner(System.in);

System.out.println("Ola!");

System.out.println("Vamos comecar:");

System.out.println("Entrando no multicast...");

MulticastSocket multicast = new MulticastSocket(processo.multicastPORT);

processo.group = InetAddress.getByName(processo.multicastAddres);

multicast.joinGroup(processo.group);

// Comeca a escutar multicast

Thread listenerThread = new Thread(){

@Override

public void run(){

processo.listenMulticast(multicast);

}

};

listenerThread.start();

System.out.println("Entramos!");

boolean running = true;

while(running) {

String opt = scanner.nextLine();

DatagramPacket data = new DatagramPacket(opt.getBytes(),

opt.length(),

processo.group,

processo.multicastPORT);

multicast.send(data);

if("ok".equals(opt)){

System.out.println("Estou saindo amigos");

multicast.leaveGroup(processo.group);

multicast.close();

running = false;

}

}

System.out.println("Encerrando processo...");

processo.running = false;

multicast.close();

scanner.close();

System.out.println("Tchau!");

}

static void main() {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

}

final class Processo {

public String multicastAddres = "230.0.0.0";

public int multicastPORT = 6789;

public InetAddress group;

final static String CRLF = ",";

public ArrayList<String> membros = new ArrayList<String>();

public boolean running = true;

public void listenMulticast(MulticastSocket multicast){

try{

while(this.running) {

byte[] buffer = new byte[10\*1024];

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

multicast.receive(data);

String linha = new String(buffer, 0, data.getLength());

String[] mensagem = linha.split(CRLF);

// Trata a mensagem

System.out.println(mensagem[0]);

}

}catch(Exception e){

//System.out.println("Deu pau listenMultiCast");

//System.out.println(e.toString());

}

}

}