 ****

**master ISI :**

**ingenieurie des sustemes d’information**

**Rapport de Tp 7 Java AVANCEE :**

**Programmation Réseau avec les sockets**

Réalisé par :

ESSADEQ Ayoub

**2022-2023**

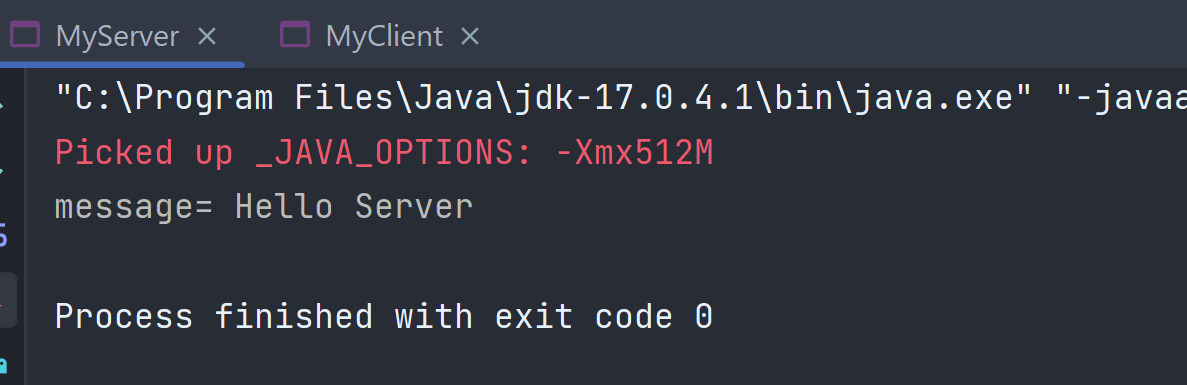
**Class myserveur :**

*import* java.io.\*;  
*import* java.net.\*;  
  
*public class* MyServer {  
  
 *public static void* main(String[] args) {  
 *try* {  
 ServerSocket ss = *new* ServerSocket(6666);  
 Socket s = ss.accept();*//establishes connection* DataInputStream dis = *new* DataInputStream(s.getInputStream());  
 String str = (String) dis.readUTF();  
 System.out.println("message= " + str);  
 ss.close();  
 } *catch* (Exception e) {  
 System.out.println(e);  
 }  
 }  
}

**Class myClient :**

*import* java.io.\*;  
*import* java.net.\*;  
  
*public class* MyClient {  
  
 *public static void* main(String[] args) {  
 *try* {  
 Socket s = *new* Socket("localhost", 6666);  
 DataOutputStream dout = *new* DataOutputStream(s.getOutputStream());  
 dout.writeUTF("Hello Server");  
 dout.flush();  
 dout.close();  
 s.close();  
 } *catch* (Exception e) {  
 System.out.println(e);  
 }  
 }  
}

on run le serveur puis la class client :



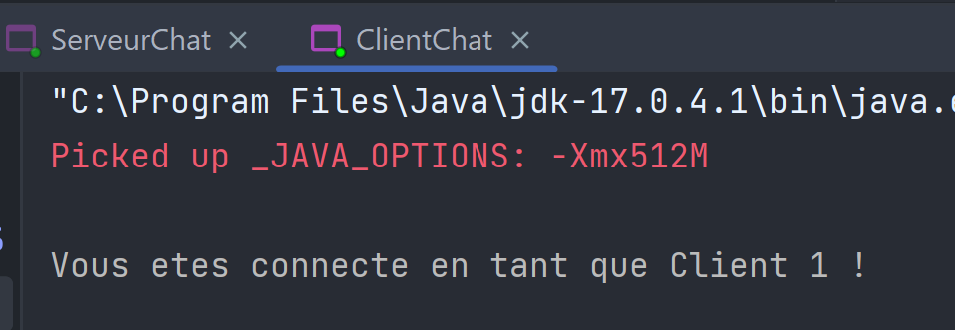
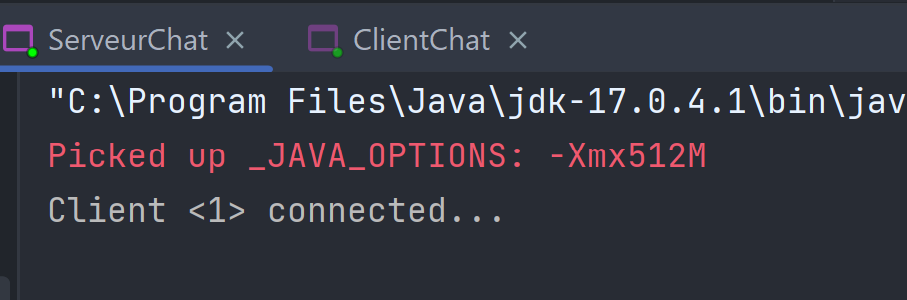
**La class serveurchat :**

*import* java.io.IOException;  
*import* java.net.ServerSocket;  
*import* java.net.Socket;  
*import* java.util.ArrayList;  
  
  
*public class* ServeurChat{  
  
 *public static void* main(String[] args)  
 {  
 ArrayList<Socket> listClientsConnectee = *new* ArrayList<Socket>(); *// liste de tous les clients qui ont acced�s au serveur  
 int* numClient = 0 ; *// nb de clients actually connected to the server* ServerSocket serv\_sock; *// gestionnaire de sockets  
  
 try* {  
 serv\_sock = *new* ServerSocket(9000);  
  
 *while*(*true*)  
 {  
 Socket sock\_client = serv\_sock.accept();  
 listClientsConnectee.add(sock\_client);  
 numClient++;  
 *new* Chat(numClient, sock\_client, listClientsConnectee).start();  
  
 }  
  
  
 } *catch* (IOException e) {  
 e.printStackTrace();  
  
 }  
  
 }  
}

**La class Clientchat :**

*import* java.io.BufferedReader;  
*import* java.io.IOException;  
*import* java.io.InputStream;  
*import* java.io.InputStreamReader;  
*import* java.io.OutputStream;  
*import* java.io.PrintWriter;  
*import* java.net.Socket;  
*import* java.net.UnknownHostException;  
*import* java.util.Scanner;  
  
*public class* ClientChat {  
  
 *private static* Socket sock\_client;  
  
 *public static void* main(String[] args) {  
  
 *try* {  
 *//creation du socket client* sock\_client = *new* Socket("localhost", 9000);  
  
 *//recuperation du flux de lecture avec le serveur* InputStream is = sock\_client.getInputStream();  
 InputStreamReader ipsr = *new* InputStreamReader(is);  
 *final* BufferedReader reader = *new* BufferedReader(ipsr);  
  
 *// recuperation du flux d'ecriture avec le serveur* OutputStream os = sock\_client.getOutputStream();  
 *final* PrintWriter writer = *new* PrintWriter(os, *true*);  
  
 Thread reception = *new* Thread(*new* Runnable() {  
 String msg;  
  
 *@Override  
 public void* run() {  
 *while* (*true*) {  
 *try* {  
 msg = reader.readLine();  
 *if* (msg != *null*) {  
 System.out.println(msg);  
 }  
 } *catch* (IOException e) *//e.printStackTrace();* {  
 }  
 }  
 }  
 });  
  
 Thread emission = *new* Thread() {  
  
 *private* Scanner sc;  
  
 *public void* run() {  
 *while* (*true*) {  
 *// pour lire � partir de l'entr� standard* sc = *new* Scanner(System.in);  
 String str = sc.nextLine();  
  
 *//envoyer ce qui est �t� lu* writer.println(str);  
 }  
 }  
 };  
  
 emission.start();  
 reception.start();  
 } *catch* (UnknownHostException uhe) {  
 uhe.printStackTrace();  
 } *catch* (IOException ioe) {  
 ioe.printStackTrace();  
 }  
 *//System.out.println("<client> end main");* }*//main*}

on run le serveur puis la class client :

****

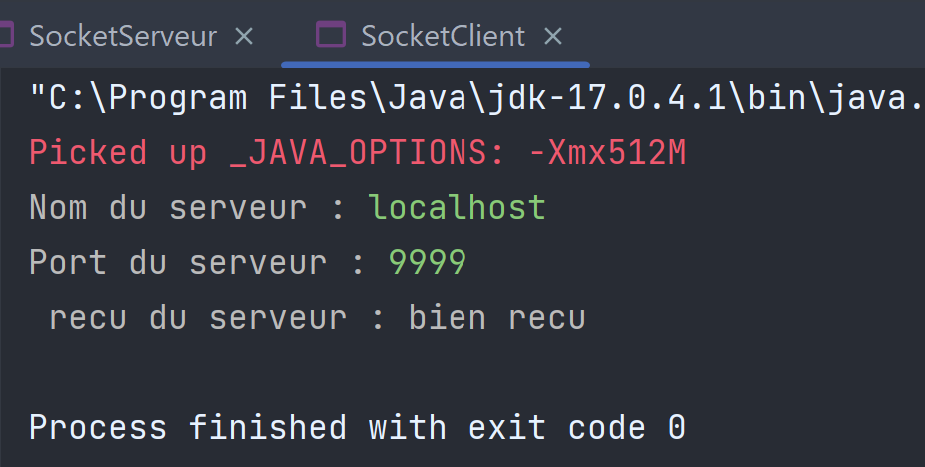
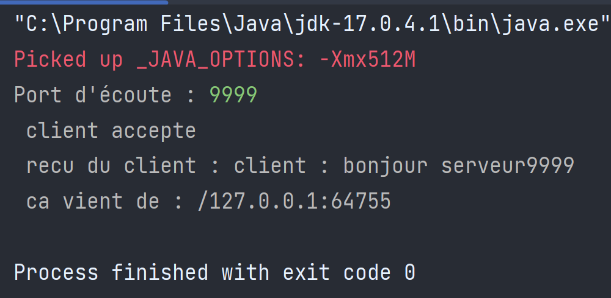
**La class SocketServeur :**

*import* java.io.\*;  
*import* java.net.\*;  
*import* java.util.Scanner;  
  
*import* java.io.\*;  
*import* java.net.\*;  
*import* java.util.Scanner;  
  
*public class* SocketServeur {  
  
 *public static void* main(String argv[]) {  
 *int* port = 0;  
 Scanner keyb = *new* Scanner(System.in);  
 System.out.print("Port d'écoute : ");  
 *try* {  
 port = keyb.nextInt();  
 } *catch* (NumberFormatException e) {  
 System.err.println("Le paramètre n'est pas un entier.");  
 System.err.println("Usage : java ServeurUDP port-serveur");  
 System.exit(-1);  
 }  
 *try* {  
 ServerSocket serverSocket = *new* ServerSocket(port);  
 Socket socket = serverSocket.accept();  
 System.out.println(" client accepte");  
 ObjectOutputStream output = *new* ObjectOutputStream(socket.getOutputStream());  
 ObjectInputStream input = *new* ObjectInputStream(socket.getInputStream());  
 String chaine = (String) input.readObject();  
 System.out.println(" recu du client : " + chaine);  
 System.out.println(" ca vient de : " + socket.getInetAddress() + ":" + socket.getPort());  
 output.writeObject(*new* String("bien recu"));  
 } *catch* (Exception e) {  
 System.err.println("Erreur : " + e);  
 }  
 }  
}

**La class Socketclient :**

*import* java.io.\*;  
*import* java.net.\*;  
*import* java.util.Scanner;  
  
*class* SocketClient {  
  
 *public static void* main(String argv[]) {  
 *int* port = 0;  
 String host = "";  
 Scanner keyb = *new* Scanner(System.in);  
 System.out.print("Nom du serveur : ");  
 host = keyb.next();  
 System.out.print("Port du serveur : ");  
 *try* {  
 port = keyb.nextInt();  
 } *catch* (NumberFormatException e) {  
 System.err.println("Le second paramètre n'est pas un entier.");  
 System.exit(-1);  
 }  
 *try* {  
 InetAddress adr = InetAddress.getByName(host);  
 Socket socket = *new* Socket(adr, port);  
 ObjectOutputStream output = *new* ObjectOutputStream(socket.getOutputStream());  
 ObjectInputStream input = *new* ObjectInputStream(socket.getInputStream());  
 output.writeObject(*new* String("client : bonjour serveur"+socket.getPort()));  
 String chaine = (String) input.readObject();  
 System.out.println(" recu du serveur : " + chaine);  
 } *catch* (Exception e) {  
 System.err.println("Erreur : " + e);  
 }  
  
 }  
}

on run le serveur puis la class client :



**La class udpBaseServer\_2 :**

*import* java.io.IOException;  
*import* java.net.DatagramPacket;  
*import* java.net.DatagramSocket;  
*import* java.net.InetAddress;  
*import* java.net.SocketException;  
  
*public class* udpBaseServer\_2 {  
  
 *public static void* main(String[] args) *throws* IOException {  
 *// Step 1 : Create a socket to listen at port 1234* DatagramSocket ds = *new* DatagramSocket(1234);  
 *byte*[] receive = *new byte*[65535];  
  
 DatagramPacket DpReceive = *null*;  
 *while* (*true*) {  
  
 *// Step 2 : create a DatgramPacket to receive the data.* DpReceive = *new* DatagramPacket(receive, receive.length);  
  
 *// Step 3 : revieve the data in byte buffer.* ds.receive(DpReceive);  
  
 System.out.println("Client:-" + data(receive));  
 *if* (data(receive).toString().equals("bye")) {  
 System.out.println("Client sent bye.....EXITING");  
 *break*;  
 }  
 receive = *new byte*[65535];  
 }  
 }  
  
 *public static* StringBuilder data(*byte*[] a) {  
 *if* (a == *null*) {  
 *return null*;  
 }  
 StringBuilder ret = *new* StringBuilder();  
 *int* i = 0;  
 *while* (a[i] != 0) {  
 ret.append((*char*) a[i]);  
 i++;  
 }  
 *return* ret;  
 }  
}

**La class udpBaseclient\_2 :**

*import* java.io.IOException;  
*import* java.net.DatagramPacket;  
*import* java.net.DatagramSocket;  
*import* java.net.InetAddress;  
*import* java.util.Scanner;  
  
*public class* udpBaseClient\_2 {  
  
 *public static void* main(String args[]) *throws* IOException {  
 Scanner sc = *new* Scanner(System.in);  
  
 *// Step 1:Create the socket object for  
 // carrying the data.* DatagramSocket ds = *new* DatagramSocket();  
  
 InetAddress ip = InetAddress.getLocalHost();  
 *byte* buf[] = *null*;  
  
 *// loop while user not enters "bye"  
 while* (*true*) {  
 String inp = sc.nextLine();  
  
 *// convert the String input into the byte array.* buf = inp.getBytes();  
 *// Step 2 : Create the datagramPacket for sending  
 // the data.* DatagramPacket DpSend  
 = *new* DatagramPacket(buf, buf.length, ip, 1234);  
  
 *// Step 3 : invoke the send call to actually send  
 // the data.* ds.send(DpSend);  
  
 *// break the loop if user enters "bye"  
 if* (inp.equals("bye")) {  
 *break*;  
 }  
 }  
 }  
}

on run le serveur puis la class client :

