**TD1 : Les Sockets**

Partie 1: les sockets TCP

Question 1 :

ServeurTDQuestion1

import java.io.BufferedReader;

import java.io.IOException;

//import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.ServerSocket;

import java.net.Socket;

public class ServeurTDQuestion1 {

ServerSocket s;

ServeurTDQuestion1 () {

try { s=new ServerSocket(1234); }

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

}

void ecouter() {

while(true)

{ try {

Socket clientSocket = s.accept();

System.out.println("Nouv client");

doService(clientSocket);

clientSocket.close();

System.out.println("Le client quitte\nEn attente d'un nouv. client"); }

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

}

}

void doService(Socket clientSocket) {

try {

BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

while(true) {

String L=in.readLine();

if(L.equals("END"))

break;

L=L.toUpperCase();

System.out.println("En maj : "+L);

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

out.println(L); }

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

}

public static void main(String arg[]) {

ServeurTDQuestion1 s= new ServeurTDQuestion1();

s.ecouter(); }

}

ClientTDQuestion1

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

//import java.net.InetAddress;

import java.net.Socket;

public class ClientTDQuestion1 {

Socket S\_Client;

ClientTDQuestion1(String add, int port) {

try { S\_Client = new Socket(add,port); }

catch (IOException e) { System.out.println(e.getMessage()); }

}

void communiquer() {

try {

//pour l envoi

BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in));

PrintStream theOutputStream = new PrintStream(S\_Client.getOutputStream());

//theOutputStream.println(userInput.readLine());

//pour la reception

BufferedReader in = new BufferedReader(new InputStreamReader(S\_Client.getInputStream()));

//String L=in.readLine();

for(int i=0; i<10;i++) {

theOutputStream.println(userInput.readLine());//envoyer

String L=in.readLine();//recevoir

System.out.println("En maj : "+L);//afficher le msg recu }

theOutputStream.println("END"); }

catch (IOException e) { System.out.println(e.getMessage()); }

}

public static void main(String arg[]) {

ClientTDQuestion1 c= new ClientTDQuestion1("127.0.0.1",1234);

c.communiquer(); }

}

Question 2 :

ServeurTDQuestion2

import java.io.BufferedReader;

import java.io.IOException;

//import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.Date;

import java.util.StringTokenizer;

public class ServeurTDQuestion2 {

ServerSocket s;

ServeurTDQuestion2 () {

try { s=new ServerSocket(1234); }

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

}

void ecouter() {

//while(true) {

try {

Socket clientSocket = s.accept();

System.out.println("Je vais envoyer la date");

envoyerDate(clientSocket);

premier(clientSocket);

clientSocket.close();

//System.out.println("Le client quitte\nEn attente d'un nouv. client"); } catch (IOException e) { System.out.println("Erreur"+e.getMessage()); }

//}

}

void envoyerDate(Socket clientSocket) {

try {

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

out.println("Nous somme le "+new Date()); }

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

}

boolean estPremier(int n) {

boolean b=false;

if(n==3 || n==2)b=true;

else

for(int i=2; i<=n/2; i++) {

if (n%i!=0)

b=true;

else { b=false; break;}

}

return b; }

void afficherPremier(int n) {

if(n<=3) System.out.println("Pas de nombre premier inférieur à "+n);

else

for(int i=2; i<n; i++) {

if (estPremier(i))

System.out.println("Nombre premier < "+n+" : "+i ); }

}

int pgcd(int m, int n) {

int r;

while(m%n!=0) {

r=m%n;

m=n;

n=r; }

return n; }

boolean premierEntre(int m, int n) {

return pgcd(m,n)==1; }

void premier(Socket clientSocket) {

try {

BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

int p = Integer.parseInt(in.readLine());

//Verifier s il est premier

if(estPremier(p)) System.out.println(p+" : nombre premier");

//afficher tous les nbres premiers inférieurs à p

afficherPremier(p);

String Liste=in.readLine();

StringTokenizer s = new StringTokenizer(Liste);

int m=Integer.parseInt(s.nextToken());

int n=Integer.parseInt(s.nextToken());

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

out.println("Premier entre eux : "+premierEntre(m,n)); }

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

}

public static void main(String arg[]) {

ServeurTDQuestion2 s= new ServeurTDQuestion2();

s.ecouter() ; }

}

ClientTDQuestion2

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

//import java.net.InetAddress;

import java.net.Socket;

public class ClientTDQuestion2 {

Socket S\_Client;

ClientTDQuestion2(String add, int port) {

try {

S\_Client = new Socket(add,port); }

catch (IOException e) { System.out.println(e.getMessage()); }

}

void communiquer() {

try {

//pour la reception

BufferedReader in = new BufferedReader(new InputStreamReader(S\_Client.getInputStream()));

String date=in.readLine();//recevoir date

System.out.println(date);

//pour l envoi

System.out.println("Donner un entier positif");//je suppose qu il est tjrs + BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in)); //pour lire du clavier

PrintStream theOutputStream = new PrintStream(S\_Client.getOutputStream());

theOutputStream.println(Integer.parseInt(userInput.readLine()));

//Envoyer une liste de deux entiers >0

System.out.println("Donner une liste de deux entiers positifs separes par des espaces, exemple : 2 15 ou 2 6");

theOutputStream.println(userInput.readLine());

System.out.println(in.readLine()+"\nFIN");

} catch (IOException e) { System.out.println(e.getMessage()); }

}

public static void main(String arg[]) {

ClientTDQuestion2 c= new ClientTDQuestion2("127.0.0.1",1234);

c.communiquer(); }

}

Premier

public class Premier {

boolean estPremier(int n) {

boolean b=false;

if(n==3 || n==2)b=true;

else

for(int i=2; i<=n/2; i++) {

if (n%i!=0)

b=true;

else { b=false; break;}

}

return b; }

public static void main(String a[]) {

Premier p=new Premier();

System.*out*.println(p.estPremier(17)); }

}

PremierEntreEux

public class PremierEntreEux {

int pgcd(int m, int n) {

int r;

while(m%n!=0) {

r=m%n;

m=n;

n=r;

}

return n;

}

boolean premierEntre(int m, int n) {

return pgcd(m,n)==1;

}

public static void main(String a[]) {

PremierEntreEux p=new PremierEntreEux();

System.*out*.println(p.premierEntre(16,8));

}

}

Partie 2: les sockets UDP

Question 1 :

ServiceUDP

import java.io.IOException;

import java.net.DatagramSocket;

import java.net.InetAddress;

public class ServiceUDP {

private DatagramSocket socket;

private InetAddress adresseIP;

private int port;

public ServiceUDP(String adresseDest, int portDest) throws IOException {

adresseIP = InetAddress.getByName(adresseDest);

port = portDest;

}

public ServiceUDP(int portEcoute) throws IOException {

}

public void envoyer(String mesg) throws IOException {

}

public String recevoir() throws IOException {

String msg="";

return msg;

}

public String getAdresse() {

String adr="";

return adr;

}

}

Question 2 :

TestServer

public class TestServer {

}

TestClient

public class TestClient {

}