Université Ain Temouchent BELHADJ Bouchaib
Institut des Sciences
Département de Mathématiques et Informatique

Master1 RID Systèmes distribués

TP N°2

Soit la classe java suivante (**Etudiant.java**), elle représente les caractéristiques d'un étudiant **(son nom, sa spécialité et sa moyenne générale).**

```
import java.io.Serializable;
public class Etudiant implements Serializable{
String nom;
String specialite;
int moy;
Etudiant (String nom, String specialite, int moy) {
this.nom = nom;
this.specialite = specialite;
this.moy = moy;
}
String getNom() {
return nom;
}
public String toString() {
return "Etudiant : "+nom+" "+specialite+" : "+moy;
}
}
```

Le serveur **(ServerEtudiant.java)** contient 3 étudiants, il récupère le nom d'un étudiant (envoyé par le client), cherche cet étudiant dans son tableau et envoie l'objet étudiant correspondant au client.
Pour lire le nom, le serveur utilise **readLine** de **BufferedReader** mais pour envoyer l'objet, il doit passer par **writeObject** de **ObjectOutputStream.**

```
import java.net.*;
import java.io.*;
class ServerEtudiant {
public static void main(String args[]) {
Etudiant[ ] tabEtudiant = {new Etudiant ("A", "GL", 13),new Etudiant ("B", "RID",
12),new Etudiant ("C", "SI", 14)};
ServerSocket server = null;
try {
server = new ServerSocket(7777);
while (true) {
Socket sock = server.accept();
System.out.println("connecte");
ObjectOutputStream sockOut = new ObjectOutputStream(sock.getOutputStream());
BufferedReader sockIn = new BufferedReader(new
InputStreamReader(sock.getInputStream()));
String recu;
while ((recu = sockIn.readLine()) != null) {
System.out.println("recu :"+recu);
String nom = recu.trim();
for (int i=0; i<tabEtudiant.length; i++)
if (tabEtudiant[i].getNom().equals(nom)) {
sockOut.writeObject(tabEtudiant[i]);break; }
}
sockOut.close();
```

```
sock.close();
} } catch (IOException e) {
try {server.close();} catch (IOException e2) {}
}
}// fin main
}// fin classe
```

Le client **(ClientEtudiant.java)** utilise **println** de **PrintWriter** pour envoyer le nom de l'étudiant au serveur et récupère l'objet étudiant (envoyé par le serveur) avec **readObject()** de **ObjectInputStream.**

```
import java.io.*; import java.net.*;
public class ClientEtudiant {
public static void main(String[] args) throws IOException {
String hostName = "localhost";
String NomEtudiant = "A";
Socket sock = null;
PrintWriter sockOut = null;
ObjectInputStream sockIn = null;
try {
sock = new Socket(hostName, 7777);
sockOut = new PrintWriter(sock.getOutputStream(), true);
sockIn = new ObjectInputStream(sock.getInputStream());
} catch (UnknownHostException e) {System.err.println("host non atteignable :
"+hostName); System.exit(1); }
catch (IOException e) {System.err.println("connection impossible avec : "+hostName);
System.exit(1);}
sockOut.println(NomEtudiant); // envoyer le nom au serveur
try {
Object recu = sockIn.readObject(); // récupérer l'objet Etudiant envoyé par le serveur
if (recu == null) System.out.println("erreur de connection");
else { Etudiant etudiant = (Etudiant)recu;
System.out.println("serveur -> client : " + etudiant);
}
} catch (ClassNotFoundException e) {System.err.println("Classe inconnue : "+hostName);
System.exit(1);}
sockOut.close();
sockIn.close();
sock.close();
}
}
```

Modifier le tableau **tabEtudiant** coté serveur afin d'y avoir 9 étudiants. 3 de chaque spécialité (RID, SI et GL). Réalisez par la suite les opérations suivantes :

- Le client se connectera au serveur en envoyant une spécialité (exemple, RID). Le serveur répondra par un tableau contenant uniquement les étudiants de cette spécialité.
- Le client se connectera au serveur en envoyant un entier (entre 1 et 19). Le serveur répondra par un tableau contenant tous les étudiants ayant une moyenne supérieure à cet entier.
- Le client se connectera au serveur en envoyant le mot 'trier'. Le serveur répondra par le tableau trié de tous les étudiants selon l'ordre décroissant de leurs moyennes.