

### **Exercice 01 (12 points)**

Un fichier texte (.txt) est un document contenant une suite finie de caractères. On veut écrire un programme permettant l'échange (émission et réception) de fichiers textes entre deux PCs appartenant au même LAN.

1. Si on utilise les sockets, quel type de sockets vous proposez? et pourquoi?

**TCP(1point), transfert fiable sans perte(1point)**

2. Ecrire le programme Client/Serveur correspondant utilisant les primitives Java:

//création socket Serveur

```
ServerSocket s = new ServerSocket(port);
```

```
Socket soc = s.accept();
```

//envoi et réception sur le socket

```
BufferedReader plec= new BufferedReader(new InputStreamReader(soc.getInputStream()));
```

```
String plec.readLine(void);
```

```
PrintWriter pred = new
```

```
PrintWriter(new BufferedWriter(new OutputStreamWriter(soc.getOutputStream()),true);
```

```
void pred.println(String str);
```

//création socket Client

```
Socket socket =new Socket("adresse IP Serveur",port);
```

//fermeture du socket et des tampons

```
plec.close(); pred.close(); socket.close();
```

//Lecture d'un fichier en Java

```
File file = new File("filename.txt"); Scanner sc = new Scanner(file);
```

```
boolean sc.hasNextLine(void); String line = sc.nextLine(); sc.close();
```

//Ecriture dans un fichier

```
FileWriter fw = new FileWriter("filename.txt"); fw.write(String Line); fw.close();
```

Client (5 points)

```
Socket socket =new Socket("localhost",8888); //0.5 point
```

```
PrintWriter pred = new //0.5 point
```

```
PrintWriter(new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()),true);
```

```
File file = new File("filename.txt"); // 0.5 point
```

```
Scanner sc = new Scanner(file); //0.5 point
```

```
while (sc.hasNextLine()){ //1.5 point
```

```
String line = sc.nextLine(); //0.5 point
```

```
pred.println(line); //0.5 point
```

```
}
```

```
sc.close();pred.close(); socket.close(); //0.5 point
```

Serveur (5 points)

```
ServerSocket s = new ServerSocket(8888); //0.5 point
```

```
Socket soc = s.accept(); //0.5 point
```

```
BufferedReader plec= new BufferedReader(new InputStreamReader(soc.getInputStream()));//0.5 point
```

```

FileWriter fw = new FileWriter("filename.txt"); //0.5 point
String line= plec.readLine();
while(!line.equals("")){ //1.5 point
    fw.write(line); //0.5 point
    line= plec.readLine(); //0.5 point
}
fw.close(); plec.close(); soc.close(); //0.5 point

```

### **Exercice 02 (6 points)**

1. Ecrire les classes Java décrivant l'énoncé suivant:

L'université (ayant comme attribut 'Nom') est composée de plusieurs département (ayant aussi comme attribut 'Nom'), chacun comporte plusieurs enseignants et plusieurs étudiants. L'étudiant et l'enseignant partagent les attribut Nom, Prénom , tandis que chacun a ses propres attribut(étudiant: année, enseignant: grade).

//4 points

```

public class Universite{ //1 point
    private String Nom;
    public Departement[] Deps;
    public Universite(String N){Nom=N;Deps=new Departement[10];}
}

public class Departement{ // 1 point
    private String Nom;
    public Enseignant Ens[];
    public Etudiant Etu[];
    public Departement(String N){Nom=N;Ens=new Enseignant[10];Etu=new Etudiant[10];}
}

public class Personne{ // 1 point
    private String Nom;
    private String Prenom;
    public Personne(String N,String P){Nom=N;Prenom=P;}
}

public class Enseignant{ //0.5 point
    private String Grade;
    public Enseignant(String N,String P,String G){ super(N,P);Grade=G;}
}

public class Etudiant{// 0.5 point
    private String Annee;
    public Etudiant(String N,String P,String A){ super(N,P);Annee=A;}
}

```

2. Ecrire un programme Java ayant la méthode "main" permettant d'instancier les objets suivant:

- ❖ l'objet université de Batna2
- ❖ les deux objets: département de biologie, français

- ❖ le département de biologie dispose de deux enseignants ("bensaad", "leila", "gradeA"), ("yousfi", "ilyes", "gradeB")
- ❖ le département de biologie dispose d'un seul étudiant ("kadri", "oussama", "master1")
- ❖ le département de français dispose d'un seul enseignant ("larbi", "ahmed", "gradeC")

```

public class Programme{ // 2 points
public static void main(String[] args){
    Universite UB2 =new Universite("Universite de Batna 2"); //0.5 point
    Departement DB= new Departement("Biologie");
    Departement DF= new Departement ("Francais");
    UB2.Deps[0]=DB;
    UB2.Deps[1]=DF; //0.25
    Enseignant ES1= new Enseignant("bensaad", "leila", "gradeA"); //0.25 point
    Enseignant ES2= new Enseignant("yousfi", "ilyes", "gradeB"); //0.25 point
    Etudiant ET1= new Etudiant("kadri", "oussama", "master1"); //0.25 point
    Enseignant ES3= new Enseignant("larbi", "ahmed", "gradeC"); //0.25 point
    DB.Ens[0]=ES1;
    DB.Ens[1]=ES2;
    DB.Etu[0]=ET1;
    DF.Ens[0]=ES3;//0.25
    }
}

```

### **Exercice 03 (2 points)**

1. Expliquer le problème de persistance lié à l'approche OO.  
les objets sont volatiles et ont besoin d'être stockés en permanence //0.5 point
2. Comment s'appelle l'environnement dans lequel vivent les composants?  
le conteneur //0.5 point
3. Donner deux exemples de technologie "middleware implicite".  
EJB CCM OU .NET // 0.5 point
4. Quelle est la raison derrière la portée limitée d'une application EJB?  
blocage du protocole RMI-IIOP par le pare-feu //0.5 point