

Devoir de contrôle

2nd Semestre (SII)

Année Universitaire : 2020- 2021

Date de l'épreuve : Mardi 09/03/2020 de 09H00 à 10H00

Matière : JAVA EE

Proposée par : Dr. OMRANI Med Amine

Classe : 4^{eme}Année GL

Nom et Prénom :

Nombre de page : 2 pages

Durée : 1^H00

Exercice 1. (8 points)

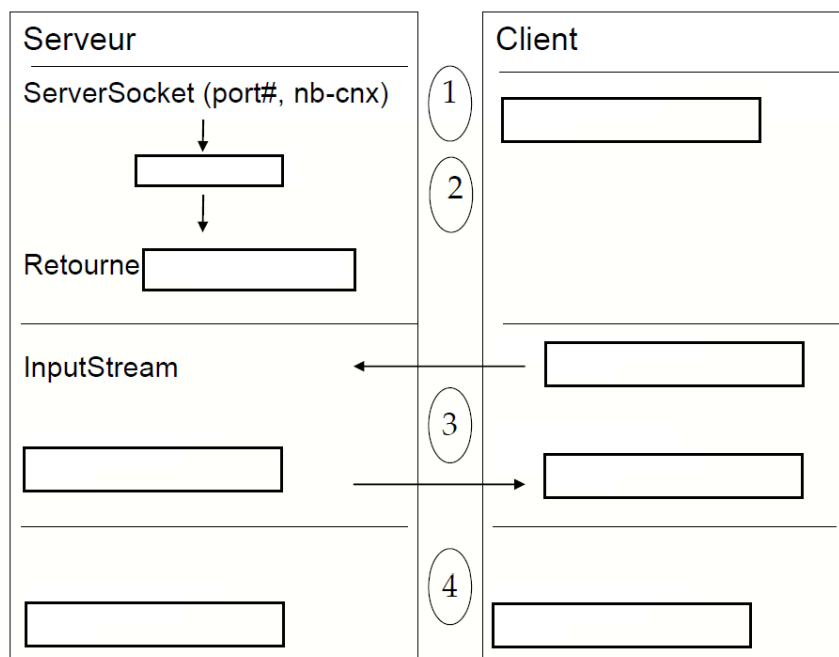
1. Définir les termes suivants (2.5 points)

JAVA EE; JSE; JSTL; JSP; Synchronized

2. Qu'est-ce qu'un socket ? Donnez les différents types de Sockets.
3. Quelle est la différence entre une communication en mode **TCP** et **UDP** ?
4. En Java, les sockets **TCP** ou **UDP** appartiennent à quel package ?
5. Un **Verrou** est associé à un objet selon deux mécanismes. Expliquez ces deux mécanismes au moyen d'un exemple.
6. Donner les deux classes essentielles pour programmer en **TCP** une communication client / Serveur ?
7. Donner les deux classes essentielles pour programmer en **UDP** une communication client / Serveur ?
8. Donner les deux classes essentielles pour **diffuser** des paquets en **UDP** une communication client / Serveur ?

Exercice 2. (6 points)

Compléter le schéma suivant illustrant le principe de fonctionnement d'un modèle Client/serveur TCP



Exercice 3. (6 points)

Soi le code suivant d'un Serveur en mode TCP/IP utilisant PrintWriter et permettant de traiter un seul client à la fois :

```
import java.net.*; import java.io.*;
class Serveur {
public static void main (String args[ ] )
{
    try
    {
        ServerSocket ss; // Socket Serveur
        Socket sc; // Socket Client

        // Enregistrer le service sur le port 5000
        // avec 5 demandes de connexion simultanées
        ss = new ServerSocket (5000, 5);

        while (true)
        { // Attendre la connexion du client et récupérer la socket
            sc = ss.accept ( );
            // Extraire un Stream d'écriture de la socket
            String message = « Message de la part du serveur »);
            OutputStream out= sc.getOutputStream ( );
            PrintWriter fout = new PrintWriter (new
                OutputStreamWriter (out));
            fout.println (message);
            sc.close ( ); // Fermer la connexion
        }
        catch (IOException e ) {System.out.println (e)};
    }
}
```

Question : Apportez les modifications nécessaires pour paralléliser le comportement de ce serveur afin qu'il puisse traiter plusieurs clients en parallèle.