

## TD PAC - Java n° 10

## Applications en mode Client-Serveur

**Exercice 1** On désire réaliser une application de type **Serveur de fichiers** en mode client-serveur.

### 1. Le Serveur

Pour cela on va créer une classe **Serveur** qui possèdera un attribut de type **ServerSocket** qui sera initialisé dans le constructeur pour attendre des connexions sur un numéro de port, éventuellement passé en ligne de commande (sinon on prendra comme port par défaut : 3456).

Lors de la connexion d'un client, le serveur devra créer un nouveau *thread* pour traiter les requêtes en provenance de ce client. Bien sûr, pendant ce temps, le serveur devra continuer d'attendre de nouvelles connexion éventuelles.

### 2. La gestion des connexions

Nous considérerons ici qu'une requête est simplement une chaîne de caractères qui correspond

- soit à un nom d'un fichier à transférer (résidant sur la machine où tourne le serveur)
- soit à la commande **fin** qui provoque l'arrêt du serveur.

Le thread traitant les requêtes sera implémenté par la classe **Connexion**. Son constructeur est appelé avec le **Socket** issu du **accept** qu'a fait le serveur. Sa méthode **run** doit lire la requête en provenance du client, la traiter, fermer la connexion, puis se terminer.

### 3. Le Client

Pour tester notre serveur de fichiers, on va écrire un autre programme qui va demander au serveur de lui envoyer un fichier. Ce programme prend en argument (dans la ligne de commande)

- le nom de la machine correspondant au serveur
- le numéro de port sur lequel se connecter
- la requête (i.e. le nom du fichier à transférer ou bien "**fin**")

Il est implémenté avec la classe **Client** qui utilisera un objet de type **Socket** pour se connecter au serveur. Le fichier récupéré est enregistré localement sous le même nom que celui utilisé dans la requête.