TD 3 sockets : Un serveur qui met des chaînes à l'envers

Objectifs :

Parallélisme via les threads, sans concurrence (ex.1), avec concurrence (ex.2).
Client/serveur.
Communication par sockets.

Exercice 1 : connexion et service

Dans cet exercice, vous devrez réaliser votre première application client/serveur. L'idée est de fournir sur le port 1234 du serveur le service d'inversion de chaîne de caractères. Le client qui se connecte envoie une chaîne de caractère, le serveur l'inverse et renvoie la chaîne inversée. La classe java.lang.StringBuffer permet avec la méthode reverse() de faire ce travail (cf javadoc).

La chaîne peut être le résultat d'une saisie clavier ou, dans un premier temps, codée en dur dans le main du client.

Exemple de traces dans les consoles (en italique les saisies clavier) :

Coté serveur : Serveur lancé sur le port 1234 Connexion 1 démarrée Connexion 2 démarrée Connexion 1 a reçu bonjour Connexion 1 terminée	**Coté client 1**: Connecté au serveur appserv.brette.fr port 1234 Tapez une chaine de caractères ➢ *bonjour* Votre chaîne inversée ruojnob	**Coté client 2**: Connecté au serveur appserv.brette.fr port 1234 Tapez une chaine de caractères ➢ _____

N'oubliez pas que le serveur doit pouvoir gérer plusieurs clients en parallèle.

Remarques :

Vous pouvez vous contenter **du coté client** d'une classe (celle de l'application) avec son main.

Du coté serveur, il vous faut une classe pour le serveur (Serveur) et une pour le service (ServiceInversion) en plus de la classe de l'application. Le main se contentera d'instancier Serveur au port 1234.

Les applications serveur et client devront être développées dans 2 projets différents, l'idée étant de pouvoir les déployer séparément.

N'oubliez pas de fermer la socket des deux cotés, la connexion pouvant être interrompue d'un coté ou de l'autre.

Exercice 2 : un service d'inscription aux cours du td2

Sur la base de l'exercice 1, écrire un programme serveur permettant à des clients de s'inscrire à un cours à nombre de places limitées (exercice du td 2).

Vous créerez dans le main un jeu d'essai en dur (quelques cours stockés dans une liste de cours, passerez la référence de cette liste à la classe ServiceCours via un setter static, puis lancerez le serveur au port 3000.
Le premier échange peut consister en un envoi de la liste des cours où il reste de la place (et combien de places il reste). Les données saisies coté client seront « Numéro du cours » et « Nombre de places demandées ». Le message renvoyé au client lui indiquera le résultat de sa requête.

En option : on pourra envisager, si le nombre de places demandées excède le nombre de places disponibles, de demander au client s'il souhaite réserver le nombre maximum qu'il reste.