

Réalisation d'une application multitâche de type « client / serveur »

Objectifs :

Développer une application multitâche en Java.

Les messages échangés peuvent être : privés, publics ou partagés dans un groupe.

Illustration sur un cas concret des problématiques liées à :

- La synchronisation d'une application client / serveur,
- L'exclusion mutuelle
- L'interblocage de processus

Prérequis :

Programmation en Java

Manipulation de la classe Thread

Manipulation de la classe ServerSocket

Description :

L'application à réaliser doit contenir :

- Un serveur composé de plusieurs threads, entre autres :
 - o *ClientListener* qui est à l'écoute des messages entrants.
 - o *ClientSender* qui envoie les messages au client.
- Un client composé de 2 tâches :
 - o *Listener* qui retourne sur la sortie standard (ou une interface graphique) tous les messages en provenance du serveur.
 - o *Sender* qui lit les messages en provenance de l'entrée standard (ou une interface graphique) et les envoie au serveur.

Remarque :

Afin de libérer le serveur pour accepter d'autres clients, il faut créer une instance de *ClientListener* ainsi qu'une instance de *ClientSender* pour chaque nouveau client accepté.

Réalisation :

Pensez à identifier les éléments à :

- Synchroniser : ***Synchronized***
- Mettre en attente : ***wait()***
- Libérer : ***notify()***, ***notifyall()***
- Interrompre : ***interrupt()***, ***isInterrupted()***
- Rendre persistant : ***setDaemon()***

Notation :

Mise en œuvre des tâches serveurs /7 (Dispatcher /4)

Mise en place de la communication inter-processus /3

Mise en place de la communication par socket /2

Mise en œuvre d'un client /2

Application (couche métier) /2

Compte Rendu /4

BONUS :

Mise en œuvre d'une interface graphique (1,5)

Mise en œuvre d'un protocole de communication (1,5)

