

TP RMI :

Dans les deux versions des chat RMI les interfaces sont placés dans un paquet « common » pour limiter les erreurs. (Java RMI class com.sun.proxy...)

La version pollthread est peu fonctionnelle, en effet le client reçoit les messages des autres clients uniquement après avoir envoyé ces propres messages. Par ailleurs, le client reçoit dans la liste des messages ses propres messages.

Le serveur contient une liste de messages, et chaque client contient également une liste de messages.

Le client exécute simplement le thread. Le thread commence par récupérer la saisie du client et l'envoie au serveur. S'il rentre le message « quit » le client s'arrête. Le thread compare ensuite la taille des listes de message côté serveur et côté client. Si la liste de message côté client est plus petite que celle côté serveur. Alors on ajoute dans les messages côté client, les messages côté serveurs manquant et on les affiche en parallèle dans la console.

La version CallBack permet à divers clients de se connecter et de recevoir les messages des clients en temps réel sans envoyer de message. Un client peut quitter le chat s'il utilise le message « quit ». Dans la classe serveur, il y a une liste de clients. La méthode ajouteMessage permet d'ajouter un client à la liste. La méthode envoieMessage permet d'afficher un message pour tous les clients de la liste.

Dans le main du client, on créer un client et on l'ajoute à la liste des clients du serveur.

Le client contient un scanner une boucle while, le scanner récupère la saisie du client, tant que celle-ci est différente de « quit », le serveur affiche le message du client à l'aide de la méthode envoieMessage en passant par le serveur.

