

TP serveur echo

Le but de ce TP est simplement d'écrire un petit serveur réalisant l'écho de ce qui lui est envoyé, en utilisant des sockets TCP en Java avec Eclipse.

Sur la page web du cours vous trouverez :

- Le répertoire **serveur-echo.zip** à décompresser dans votre répertoire **eclipse-workspace**
- Un serveur echo déjà réalisé **serveur-echo.jar** à télécharger sur votre machine si ce n'est déjà fait (répertoire de votre choix)
- Le support de cours sur le modèle client-serveur est également disponible

Pour tester le serveur :

1. Lancez l'exécution du fichier serveur-echo.jar choisissez un port de n° supérieur 1024 et cliquez sur le bouton **démarrer**.

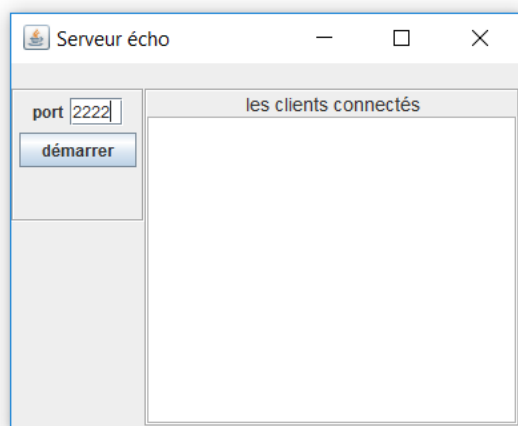


Figure 1 : démarrage du serveur

2. Lancez l'exécution du client générique **client.jar** que vous avez téléchargé à la séance précédente.
3. Vérifiez le bon fonctionnement du serveur en vous connectant sur votre serveur (machine localhost) et même port que votre serveur.
4. Envoyez du texte au serveur, il doit vous répondre en vous renvoyant le même texte.

Dans l'exemple ci-dessous (figure 2) on voit 2 clients connectés au serveur.

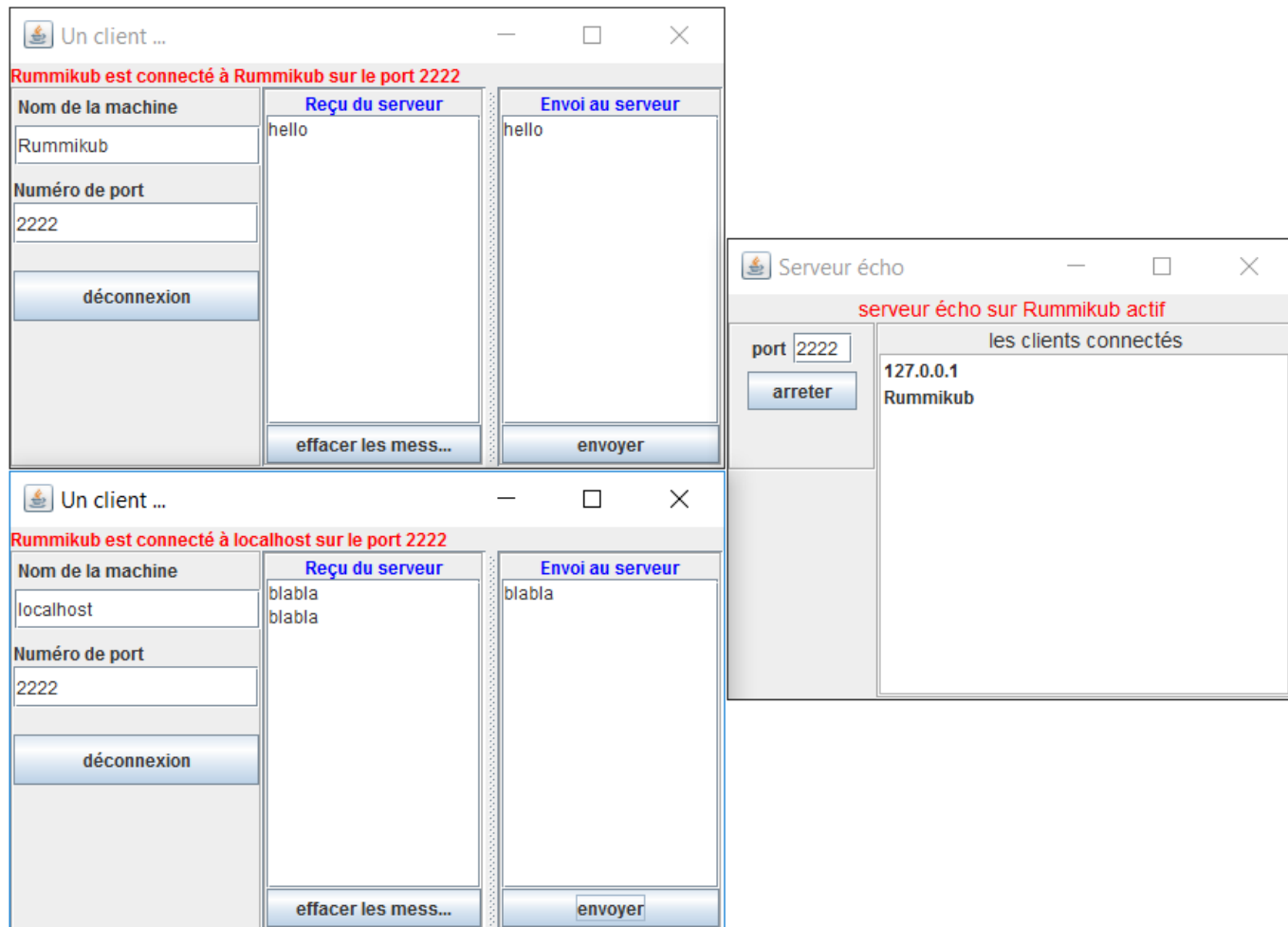


Figure 2 : Exemple de connexion de 2 clients au serveur

Création d'un serveur echo avec Eclipse

Importez dans Eclipse le projet déjà existant dans votre espace de travail (sélectionnez le répertoire `p:\workspace\serveur-echo`). L'importation faite, lancez l'exécution du programme. Vous obtenez l'interface suivante sans les parties réseau que vous avez à programmer :

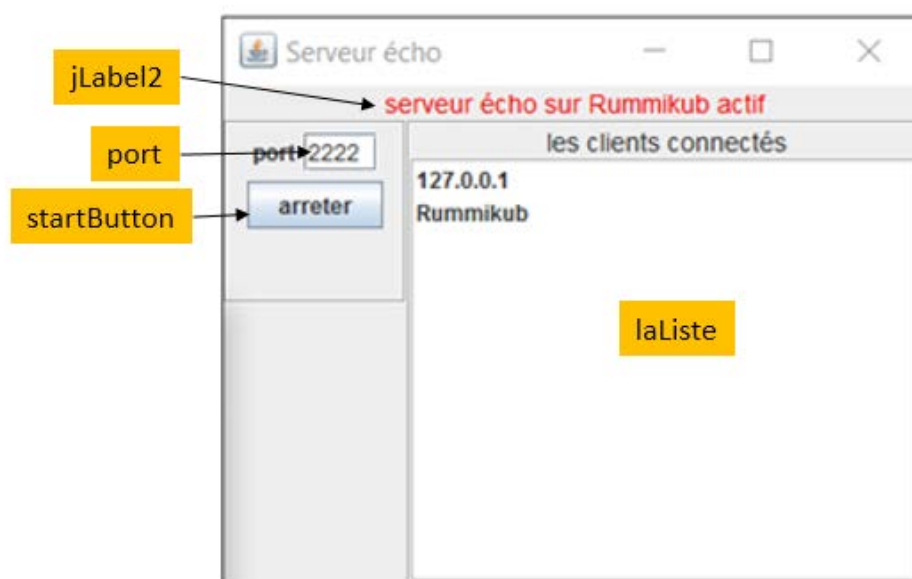


Figure 3 : Variables utilisées pour désigner les différents composants de l'interface utilisateur

L'application est composée de trois fichiers :

- AEchoServeur.java : le programme principal implémentant l'interface utilisateur,
- Serveur.java : le thread java implémentant le modèle veilleur-exécutant
- Ecoute.java : le thread java dialoguant avec un client

Travail demandé :

Programmer les parties de code manquantes de Serveur.java, Ecoute.java et le code du bouton Démarrer/Arrêter, c'est-à-dire la méthode **startButton_actionPerformed**