

PROJET

Nous avons choisi de compléter notre projet des tps précédents afin de créer une librairie de fonction au fur et à mesure des tps.

Le serveur va lancer un thread pour chaque nouveau client avec toutes ses variables qui seront partagées.

Architecture :

- Advert : objet représentant une annonce qui contient un id, un contenu et l'identifiant du vendeur.
- Client : permet de vérifier les demandes de l'utilisateur faites au clavier, les envoyer au serveur et recevoir ce que nous envoie le serveur.
- Message: énumération des différents messages possibles.
- MessageServer : envoie les messages par le serveurs en fonction du protocole
- MyException : définition des différentes erreurs possibles dans les messages reçus.
- Server : objet représentant le serveur avec une socket, une liste d'annonces, une liste d'utilisateur, un compteur pour les id utilisateurs et un pour les id des annonces.
- ServerThread : objet représentant un client qui est connecté au serveur, reçoit les messages, les vérifie et envoi un message au client une réponse.
 - bufferedReader pour lire l'input de la socket.
 - messageServer pour écrire sur l'ouput de la socket.
 - socket pour conserver la socket sur laquelle le serveur nous a accepté
 - boolean pour savoir si l'utilisateur se déconnecte.
 - hashMap d'annonce pour avoir toutes les annonces : variable partagé.
 - hashMap d'utilisateur pour avoir toutes les utilisateurs : variable partagé.
 - counter pour savoir l'id d'une nouvelle annonce et nouveau user : variable partagé.
 - idUser : id de l'utilisateur de ce thread.
- User : objet représentant un utilisateur qui contient un id, une adresse, un port et une liste de ses annonces.

Nous avons choisi d'utiliser des HashMap pour stocker nos éléments car on peut les supprimer en $O(1)$ (éviter de parcourir la structure à chaque fois).