

Au cours de ce TP nous allons reprendre l'exemple d'un client de chat qui se connecte au serveur du précédent TP en utilisant des threads.

Le code du TP précédent est disponible sur Eprel.

1 Classe Client

Question 1 Créer la classe Client, ayant pour membre privée

- Un objet de la classe SocketChannel
- Un booléen

Écrire le constructeur prenant en entrée une adresse internet et un numéro de port et qui établit la connexion sur cette adresse et ce numéro. Le booléen est initialisé à **true**. *Utiliser la classe InetSocketAddress()*.

Question 2 Ajouter un *getter* pour le booléen appelé **public** bool isConnected() et un setter.

2 Classe RepeatNetwork

Question 3 Écrire une classe RepeatNetwork ayant pour membre privé.

- Un objet de la classe SocketChannel
- Un objet de la classe ByteBuffer
- Un objet de la classe Client

Écrire un constructeur, prenant en argument un SocketChannel et un Client. et qui initialise le ByteBuffer.

Question 4 Rendre la classe compatible avec l'interface Runnable en implémentant la fonction **void** run() qui imprime tous les messages pouvant être lu sur le SocketChannel à l'écran, tant que l'objet Client est connecté. En cas de problèmes le client est marqué comme déconnecté. *Utiliser les méthodes du Socket et du ByteBuffer.*

Question 5 Tester votre code en appelant la méthode run() de cette classe depuis Client.

3 Classe RepeatKeyboard

Question 6 Idem question 3 avec un champ supplémentaire de type ReadableByteChannel initialisé grâce à Channels.newChannel(System.in)

Question 7 Rendre la classe compatible avec l'interface Runnable en implémentant la fonction **void** run() qui recopie tout les messages pouvant être lus sur clavier dans le socket. *Utiliser les méthodes du Socket, du ByteBuffer et du ReadableByteChannel.*

4 Classe Client 2

Question 8 Ajouté une méthode start () à la classe Client qui fabrique deux threads à partir des classes RepeatNetwork et RepeatKeyboard. Démarrer les deux threads et attendre que les threads finissent de s'exécuter.

Question 9 Ajouté un mécanisme en modifiant le serveur et le client pour que le client puisse ajouter un pseudonyme qui est affiché sur tous ses messages.

Question 10 Ajouté un mécanisme pour envoyer des messages privés entre les clients sans passer par le serveur.