

Au cours de ce TP nous allons reprendre l'exemple de serveur de chat du cours.

1 Serveur

Question 1 Créer la classe `Server` ayant pour membre

- Un objet de la class `ServerSocketChannel`
- Un objet de la class `Selector`

Écrire le constructeur prenant en entrée un numéro de port et qui “bind” le socket du serveur au numéro de port.

Utiliser la classe `InetSocketAddress`.

Question 2 Écrire une méthode “`void accept()`”, qui accepte un nouveau client et ajoute son socket dans le selecteur pour les événements de lecture.

Utiliser les méthodes de `SocketChannel`

Question 3 Écrire une méthode “`void repeat(SelectionKey)`” vide pour l'instant. Écrire une méthode qui contient la boucle principale du serveur qui utilise les deux méthodes précédentes pour accepter et retransmettre les messages. Ici en pseudo code pour le cas général.

```
openConnections();
Set s = {all socket waiting for I/O}
while(true){
    Set s2 = select(s, timeout);
    for( c : s2 ){
        readWriteAccept(c);
    }
}
```

Utiliser les classes `Selector` et `SelectionKey`

Question 4 Remplacer le corps de la méthode “repeat” par

```
System.out.println( "Nouveau message" );
```

Écrire un main qui lance le serveur.

Tester votre code.

- Avec `netcat(nc)` sous linux et MacOS
- Avec `telnet` dans CMD sous Windows

Question 5 Réécrire la méthode “repeat” pour qu'elle retransmette le message reçu à tous les clients.

Utiliser les classes `ByteBuffer`, `SelectionKey`, `SocketChannel`

Question 6 Tester votre code !