

## Exo1

1/

La classe `MulticastSocket` en Java est utilisée pour envoyer et recevoir des paquets IP multicast. Elle permet la communication avec des groupes multicast en utilisant le protocole UDP (User Datagram Protocol). Cette classe étend la classe `DatagramSocket` et offre des fonctionnalités spécifiques pour la gestion des communications multicast.

- Création d'un socket multicast : La classe permet de créer un socket multicast en spécifiant le port souhaité.
- Rejoindre un groupe multicast : Elle offre la méthode `joinGroup(InetAddress mcastaddr)` pour rejoindre un groupe multicast spécifique.
- Envoi de messages : Permet d'envoyer des datagrammes à un groupe multicast donné, où tous les membres du groupe recevront le message.
- Réception de messages : Les sockets multicast peuvent recevoir des datagrammes envoyés par d'autres hôtes au groupe/port multicast.
- Quitter un groupe multicast : La méthode `leaveGroup(InetAddress mcastaddr)` permet à un socket de quitter un groupe multicast auquel il était précédemment abonné.

2/

- Ce code implémente une communication unidirectionnelle entre un émetteur (`Emetteur.java`) et un récepteur (`Recepteur.java`) en utilisant des sockets de diffusion multicast (multicast sockets) en Java.
- Lorsque vous exécutez `Recepteur.java`, il se met en attente de réception sur le groupe multicast 230.0.0.0 au port 1000. Ensuite, lorsque vous exécutez `Emetteur.java`, il envoie le message "JE SUIS UN ETUDIANT INFORMATIQUE" sur le même groupe multicast et port. Le récepteur reçoit alors le message et l'affiche.

3/

Lorsque j'exécute le programme, il ne fonctionne pas correctement. Le problème est que l'adresse IP "200.0.0.0" n'est pas une adresse IP de diffusion multicast valide.

- Les adresses IP de diffusion multicast valides se situent dans les plages suivantes:  
224.0.0.0 à 239.255.255.255

## **Exo2**

### **1. Intérêt de DataOutputStream et DataInputStream pour UDP**

Les classes `DataOutputStream` et `DataInputStream` en Java sont utilisées pour écrire et lire des données primitives dans un flux d'octets respectivement. Dans le contexte de la communication par protocole UDP (User Datagram Protocol), ces classes offrent plusieurs avantages :

#### **a) Pour l'envoi de données (ClientUDPInt) :**

- i. Facilité d'utilisation
- ii. Sérialisation des données
- iii. Compatibilité :

#### **b) Pour la reception de données (ServeurUDPInt):**

- i. Facilité de lecture
- ii. Conversion des données
- iii. Gestion des types de données