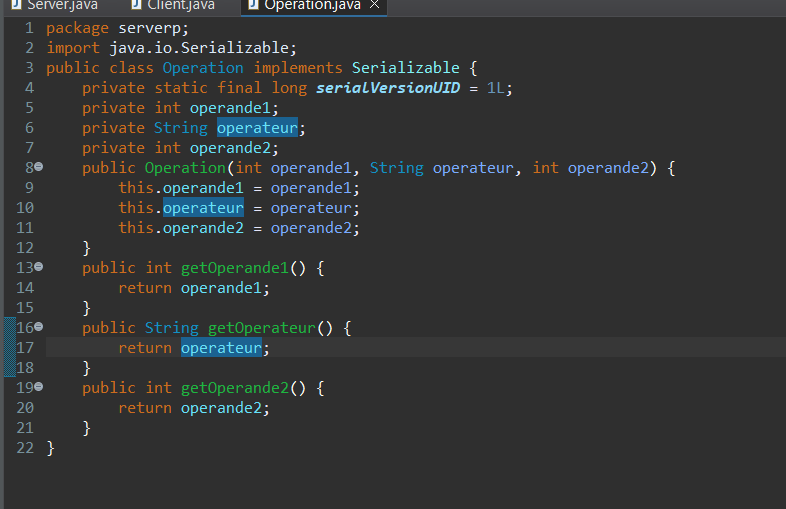
TP2.2

#### Classe Operation :

Je voulais envoyer toute l’opération dans un seul paquet, pas en trois morceaux.  
Alors, j’ai créé une **nouvelle classe appelée Operation**.  
Elle contient :

* Le premier nombre (operande1)
* L’opérateur (+, -, \*, /)
* Le deuxième nombre (operande2)

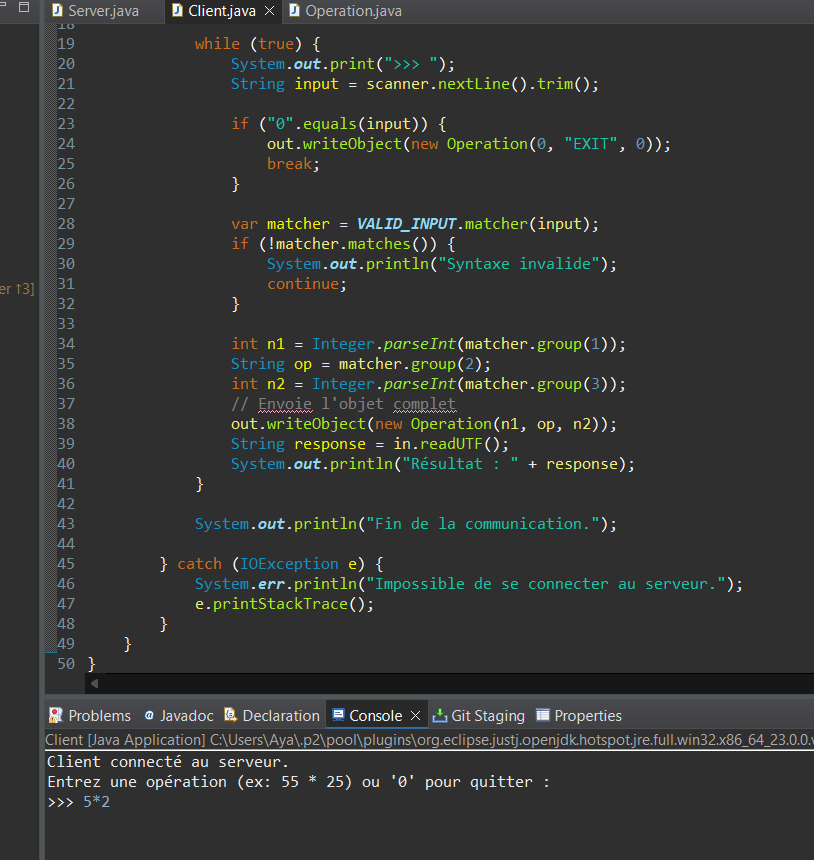
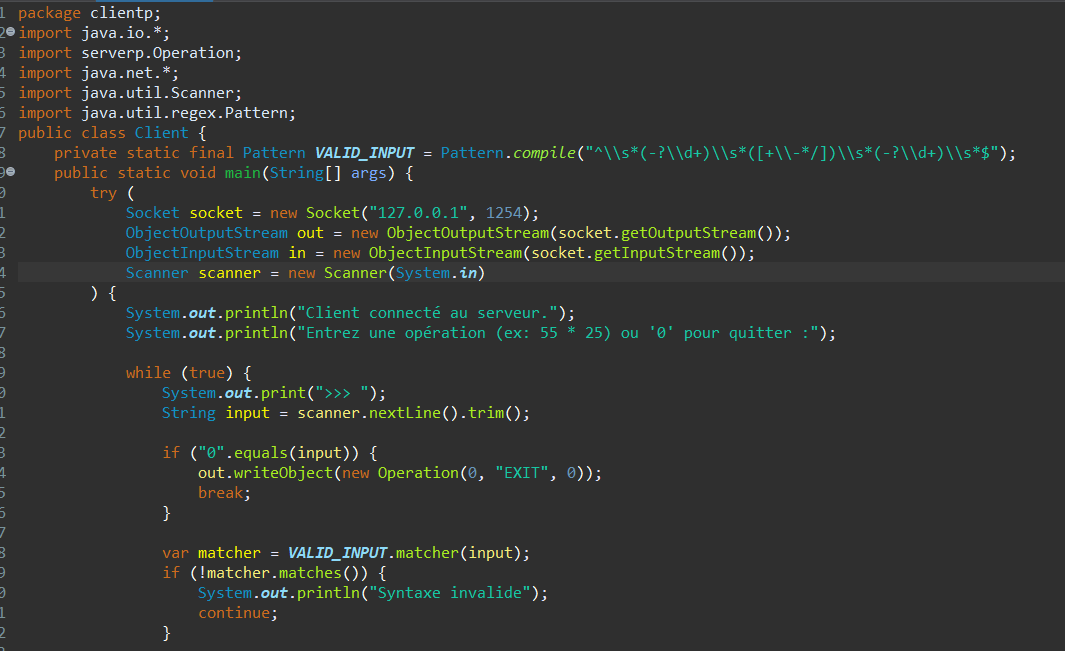
Et surtout, elle **implémente Serializable** — ça veut dire qu’on peut l’envoyer par le réseau comme un vrai objet, pas juste des nombres ou du texte.



#### Classe Client :

Avant, j’envoyais les 3 éléments un par un avec writeInt() et writeUTF().  
Maintenant, je crée un **objet Operation** avec les 3 valeurs, et je l’envoie d’un coup avec out.writeObject(...).

J’utilise aussi ObjectOutputStream et ObjectInputStream au lieu de DataOutputStream parce qu’ils savent lire /écrire des objets.



#### Classe server :

#### Le serveur, lui, reçoit maintenant un objet complet avec in.readObject(), pas 3 valeurs séparées.Il convertit cet objet en variables avec les méthodes getOperande1(), getOperateur(), etc. Ensuite, il fait le calcul (comme avant), et renvoie le résultat sous forme de texte avec out.writeUTF(...).

#### 