

TP 1 - Système Réparties

Guide d'Utilisation

Les classes envoyées côté Client sont stockées dans "src/clientFiles". Ils seront stockées par le Serveur dans "serverFiles/clientFiles".

De base il y a deux classes :

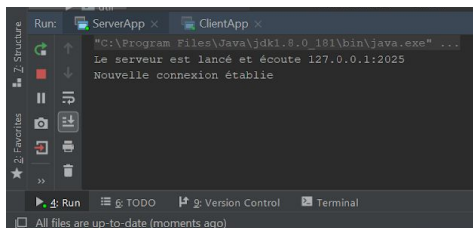
ComplexCalc			SimpleCalc		
multiply(int, int)	int		add(int, int)	int	
divide(int, int)	int		subtract(int, int)	int	

Etape 1 - Lancer le serveur

Aller dans le fichier "ServerApp.java" dans le dossier "src" et lancez le main.

Le serveur attend maintenant une connexion du client.

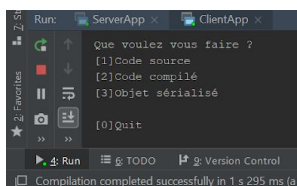
Lorsqu'un client se sera connecté, vous devriez voir apparaître le résultat suivant dans la console:



Etape 2 - Lancer le client

Aller dans le fichier "ClientApp.java" dans le dossier "src" et lancez le main.

Vous devriez maintenant avoir le résultat suivant dans votre console:

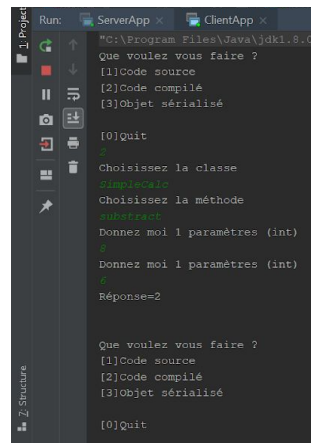


Etape 3 - Choix du protocole

Tapez le code associé au protocole.

Vous devez maintenant entrer le nom la classe à tester. Pour l'exemple nous allons prendre la classe "SimpleCalc". Vous ne devez pas écrire l'extension du fichier. Ensuite vous devez entrer la méthode à tester.

Vous devez maintenant taper un à un les deux paramètres. Lorsque vous avez entré les deux paramètres vous devriez avoir un résultat semblable à la photo ci-joint. Le résultat est affiché dans votre console. Vous pouvez choisir un autre protocole et une autre classe à exécuter ou entrer 0 pour quitter l'application.



```
Run: ServerApp x ClientApp x
"C:\Program Files\Java\jdk1.8.0_
Que voulez vous faire ?
[1]Code source
[2]Code compilé
[3]Objet sérialisé
[0]Quit
Choisissez la classe
SimpleCalc
Choisissez la méthode
subtract
Donnez moi 1 paramètres (int)
1
Donnez moi 1 paramètres (int)
2
Réponse=2

Que voulez vous faire ?
[1]Code source
[2]Code compilé
[3]Objet sérialisé
[0]Quit
```

J'ai également créer une classe pour pouvoir tester qui s'appelle "TestClient". Si vous lancez son main, il testera les trois protocoles.

Explication

Une fois le choix de protocole rentré, le Client va envoyer son choix au serveur. Le serveur renvoie un message "ACK" au "Client".

Pour le protocole ByteColl et SourceColl, nous devons envoyer le fichier. Tout d'abord le Client envoyons le nom du fichier et sa taille. Le serveur renvoie alors deux messages "ACK": un pour le nom et la taille du fichier et un pour signaler que le fichier a bien été reçu.

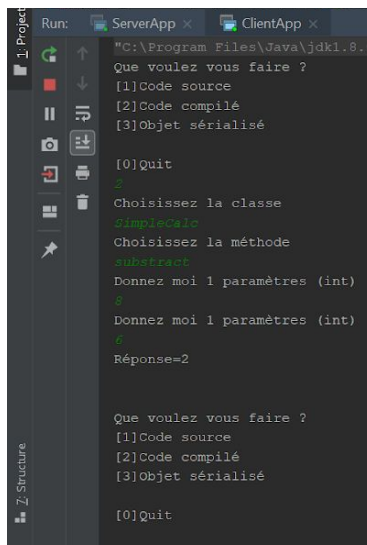
Ensuite le Client envoie séparément le nom de la méthode, le premier paramètre et le second paramètre.

Le serveur envoie alors le résultat de la méthode.

Pour le protocole ObjectColl, le Client envoie directement l'objet au serveur qui renvoie un message "ACK" une fois la réception de l'objet finie. Ensuite le Client envoie séparément le nom de la méthode, le premier et le second argument. Il recevra ensuite le résultat et la connexion se fermera.

Trace

ByteColl



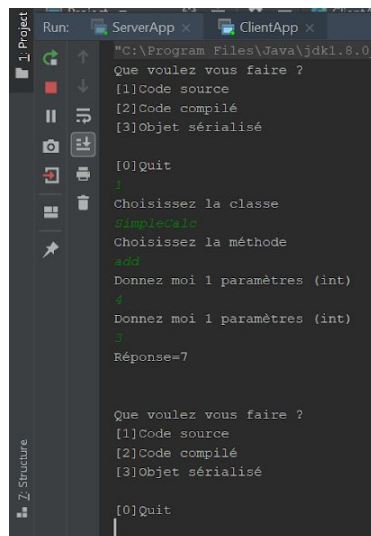
```
Run: ServerApp x ClientApp x
"C:\Program Files\Java\jdk1.8.0_101"
Que voulez vous faire ?
[1]Code source
[2]Code compilé
[3]Objet sérialisé

[0]Quit
3
Choisissez la classe
SimpleCalc
Choisissez la méthode
subtract
Donnez moi 1 paramètres (int)
8
Donnez moi 1 paramètres (int)
6
Réponse=2

Que voulez vous faire ?
[1]Code source
[2]Code compilé
[3]Objet sérialisé

[0]Quit
```

SourceColl



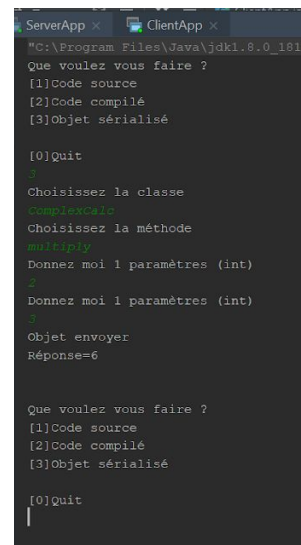
```
Run: ServerApp x ClientApp x
"C:\Program Files\Java\jdk1.8.0_101"
Que voulez vous faire ?
[1]Code source
[2]Code compilé
[3]Objet sérialisé

[0]Quit
1
Choisissez la classe
SimpleCalc
Choisissez la méthode
add
Donnez moi 1 paramètres (int)
4
Donnez moi 1 paramètres (int)
3
Réponse=7

Que voulez vous faire ?
[1]Code source
[2]Code compilé
[3]Objet sérialisé

[0]Quit
```

ObjectColl



```
Run: ServerApp x ClientApp x
"C:\Program Files\Java\jdk1.8.0_101"
Que voulez vous faire ?
[1]Code source
[2]Code compilé
[3]Objet sérialisé

[0]Quit
3
Choisissez la classe
ComplexCalc
Choisissez la méthode
multiply
Donnez moi 1 paramètres (int)
2
Donnez moi 1 paramètres (int)
3
Objet envoyer
Réponse=6

Que voulez vous faire ?
[1]Code source
[2]Code compilé
[3]Objet sérialisé

[0]Quit
```

“