

# Guide d'utilisation

La seule dépendance pour lancer le projet est le sdk de Java 1.8.

- 1- Lancer le serveur via la fonction *main* de la classe *Server*.  
Le serveur se lance automatiquement sur le port 8080 (modifiable dans la classe).
- 2- Lancer le client via la fonction *main* de la classe *Client*.  
Le client se connecte automatiquement au serveur sur le port 8080 à l'adresse 127.0.0.1
- 3- Choisir le protocole de *cloud computing* à utiliser parmi les 3 proposés : OBJECTColl (1), BYTEColl (2) et SOURCEColl (3)
- 4- Donner le dossier de travail (dossier correspondant à l'application CloudComputing).  
*exemple : /Volumes/Data/simonmarchand/Documents/projects/CloudComputing (sur MacOS)*
- 5- Donner le nom de la Class à utiliser (ici seul « Calc » a été implémenté)
- 6- Donner le nom de la méthode à utiliser (ici seul « add » a été implémenté)
- 7- Donner les différents paramètres de la méthode (ici entrer seulement 2 « int » successivement en appuyant sur « Entrer »)
- 8- Le client affiche la réponse du serveur.

# Trace d'exécution

- 1- Le serveur une fois lancée est en écoute sur le port 8080
- 2- Le client une fois lancée se connecte au serveur et le serveur lance une connexion dans un *thread* gérant ce client (il est ainsi possible de lancer plusieurs clients)
- 3- Une fois la connexion établie, le serveur attend un message (définissant une requête) du client formé de la manière suivante :

```
strategy + ''
+ filePath + ''
+ fileName + ''
+ fileExtension + ''
+ method + ''
+ parameters.get(0) + ''
+ parameters.get(1);
```
- 4- Le client réalise la création de la requête à envoyer (partie UI) et l'envoi au serveur
- 5- Le serveur reçoit le message et crée l'objet Requête puis attend l'envoi par le client de la longueur du fichier à recevoir
- 6- Le client envoie la taille du fichier
- 7- Le serveur reçoit la longueur du fichier et attend l'envoi par le client du fichier
- 8- Le client envoie le fichier dans le format demandé (.java, .class, .obj) préalablement généré :
  - a. SOURCEColl : envoi du fichier source original
  - b. BYTEColl : envoi du fichier .class généré par la méthode *compileClass()*
  - c. OBJECTColl : envoi d'un objet sérialisé
- 9- Le client est en attente de la réponse
- 10- Le serveur reçoit le fichier et le gère le fichier en fonction du protocole choisi :
  - a. SOURCEColl : réception du fichier source original, compilation du fichier, instantiation d'un objet et appel de la méthode demandée
  - b. BYTEColl : réception du fichier .class, instantiation d'un objet et appel de la méthode demandée
  - c. OBJECTColl : réception de l'objet sérialisé, désérialisation et appel de la méthode demandée
- 11- Le serveur envoie la réponse
- 12- Le client reçoit la réponse et ferme la connexion