

**Systèmes Répartis**

**Faculté des Sciences, de Génie et d’Architecture (FSGA)**

**Devoir #2**

A réaliser individuellement

La date limite de remise est fixée au **Vendredi 09 Juin 2023**

**Note1 :** Les fichiers (document en PDF, fichiers sources et exécutables) du devoir doivent être postés au portail Web du cours jusqu’au **Vendredi 09 Juin 2023 à 23h59mn59sec.** Passé ce délai, aucun devoir ne sera admis.

**Note2 :** Aucun devoir ne sera reçu par mail, la note **0.00** sera accordée automatiquement à tout étudiant n’ayant pas remis leur devoir.

**Note3 :** Le fichier Ms. Word et le code du programme doivent être compressé dans un fichier **.zip** ou **.rar** et posté au portail du cours. Pour info sur comment compresser des fichiers sur Windows, regardez cette vidéo :

Réf. <https://www.youtube.com/watch?v=hvIlEkxJPrU> )

|  |  |  |
| --- | --- | --- |
| **1** | **Nom** | **Dieudonne** |
| **2** | **Prénom** | **Sarai** |

**Exercice :** **Implémentation d’une application CHAT**

* Concevez et développez une application CHAT en utilisant la méthode « **Java Remote Method Invocation**- **Java RMI** » en JAVA.
* Votre système doit contenir :
  + Un programme Server : **myRmiJavaServer**
  + Un programme : **myRmiJavaClient**

1. **Déploiement**

**Cas 1** : Le **programme server** et le **programme client** sont déployés dans un seul environnement, c’est-à-dire sur une seule et même machine.



Pour créer cette application de Chat on a utilse 4 fichiers, 2 fichiers servant d’interface ChatServer.java, ChatClient.java et deux fichiers d’implementations ChatClientImpl.java et ChatClientImpl.java.

Dans les deux fichiers d’interface on ne fait que d’ecrire le prototype des methodes que nous allons utiliser respectivement dans chaque fichier d’implementation.

Dans le fichier ChatServerImpl.java nous avons defini plusieurs methodes telques:

La fonction RegisterClient qui prend en parametre le nom du client et l ‘interface client, servira d’ajouter chaque client connecte dans une liste prealablement defini.

La fonction broadcastMessage elle prend en parametre le nom du client qui envoie le message (String) et le message envoye(String)

Dans cette fonction on a une boucle qui va parcourir la liste mais a partir du nom des clients si le client n’est pas celui qui envoie le message il pourra recevoir le message (fonction recieve).

Dans notre methode principal a chaque execution on aura la creation d’une instance de ChatServerImpl (ou du moins une instance du serveur) etant donne qu ‘on utilise la methode de RMI, par defaut on s’en sert du port 1099, mais l’adresse pour trouver le serveur est le Localhost puisqu ‘on travaille en local.

Dans le chatClientImpl une instance de cette classe est former du nom et de l’instance du serveur.

Nous trouvons plusieurs methodes telques:

Le startChat tant que cette methode s’execute, a partir de l’instance de serveur on appelle la methode broadcast qui pourra envoyer les messages ecrits par le client et enfin la methode receive qui ne fait qu’afficher du texte.

Dans cette fonction principale On recoit le nom du client connecte

on cree une nouvelle instance a partir de ce nom et du server auquel il est connecte puis on appel la fonction startChat.

Pour compiler ces fichier d’abord on active le RMI rmi&

puis on compile les fichiers ensuite on execute le fichier d’implementation du serveur sur un terminal different du client.

Pour debuter la conversation on execute des instances de client, une fois connecte on ecrit son nom pour s’enregistrer puis le Chat peut se faire.

**Cas 2** : Le **programme server** est déployé sur **une machine virtuelle** et le programme client est déployé en même temps sur une **autre machine virtuelle**.

Dans le cas de deux machines differentes la seule difference est que dans l’adresse du serveur au lieu d’avoir le localhost on le remplace par l ‘addresse ip des machines correspondant, pour s’assurer qu’il y ait communication entre les deux machines on met le reseau sur le mode bridge ensuite pour une machine windows on s’assure que le par-feu soit desactiver.