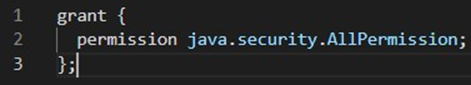
TP0 - M2 GLSI - 2022

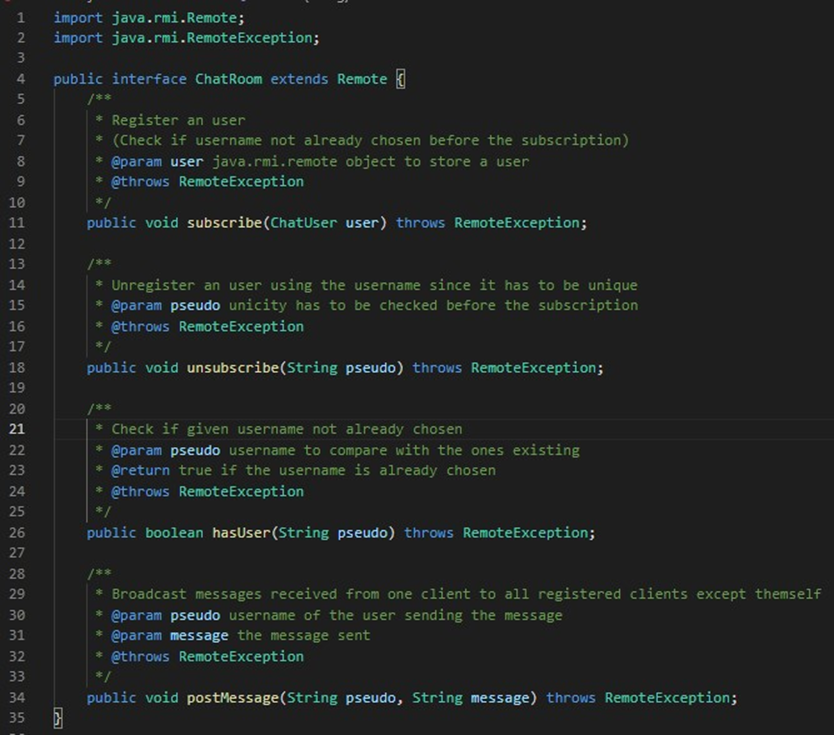
* Ahmadou Moustapha Beye
* Sall Aminata Binta Ly
* Boubou Laye Ndiaye
* Cheikh Sadibou Charles Ndiaye

Il s’agit de réaliser une chatroom en utilisant l’API Java RMI (Remote Method Invocation). Le principe sera d’avoir un serveur pour le chatroom et différents utilisateurs dont la communication (messages) transitera par ledit serveur qui se chargera de distribuer celle-ci. Le processus de réalisation est détaillé dans la suite.

Tout d’abord, afin d’avoir accès aux ressources système, il faudra accorder des permissions. Dans notre cas, ce sera fait comme suit, dans un fichier nommé ***security.policy***:



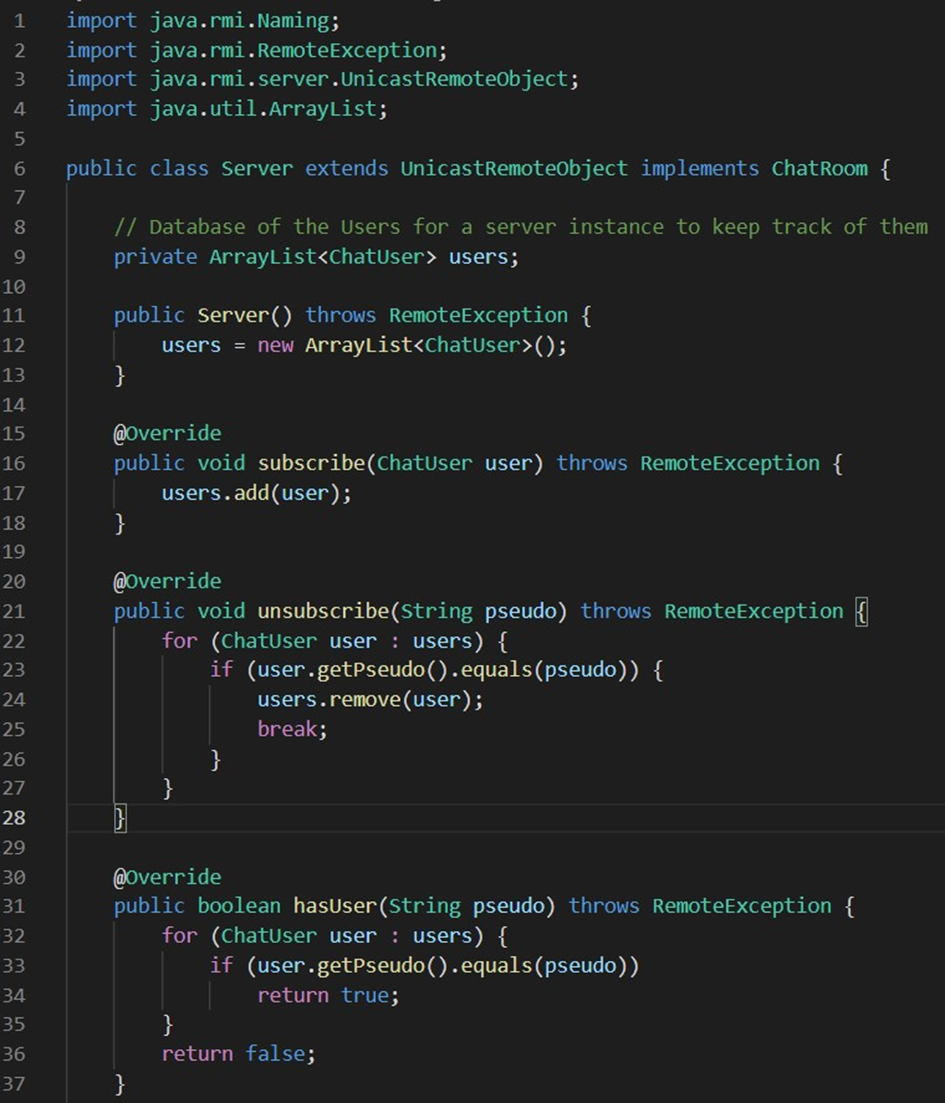
Ensuite, dans un fichier ***ChatRoom.java***, sera implémentée l’interface ***ChatRoom***, extension de ***java.rmi.Remote***.

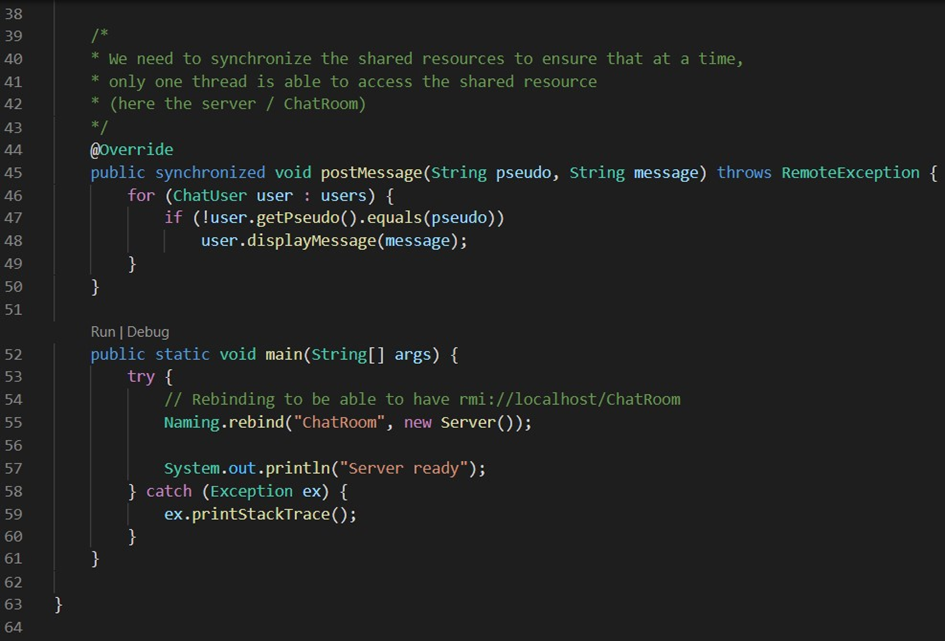


Cette interface offrira les méthodes à implémenter suivantes :

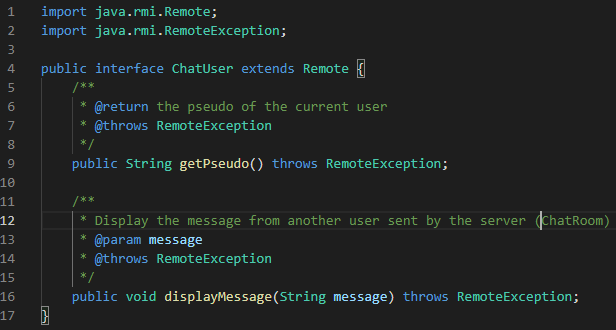
* ***subscribe*** : ne retourne pas de valeur et prend en argument un utilisateur (instance de ***ChatUser*** étendue par ***User***). Elle permet d’ajouter un utilisateur à supposer que le pseudo qui lui est associé n’existe pas déjà.
* ***unsubscribe*** : ne retourne pas de valeur et prend en argument une chaîne de caractères nommée pseudo représentant le pseudo de l’utilisateur (devant être unique). Elle supprime l’utilisateur des enregistrements d’utilisateurs.
* ***hasUser*** : retourne un booléen et prend en argument une chaîne de caractères nommée pseudo représentant le pseudo de l’utilisateur (devant être unique). Elle retourne ***true*** si le pseudo fourni existe déjà.
* ***postMessage*** : sans valeur de retour, elle diffuse le message (second paramètre), reçu d’un utilisateur (pseudo en premier paramètre), aux autres.

***Server.java*** contiendra l’implémentation de ces différentes méthodes :



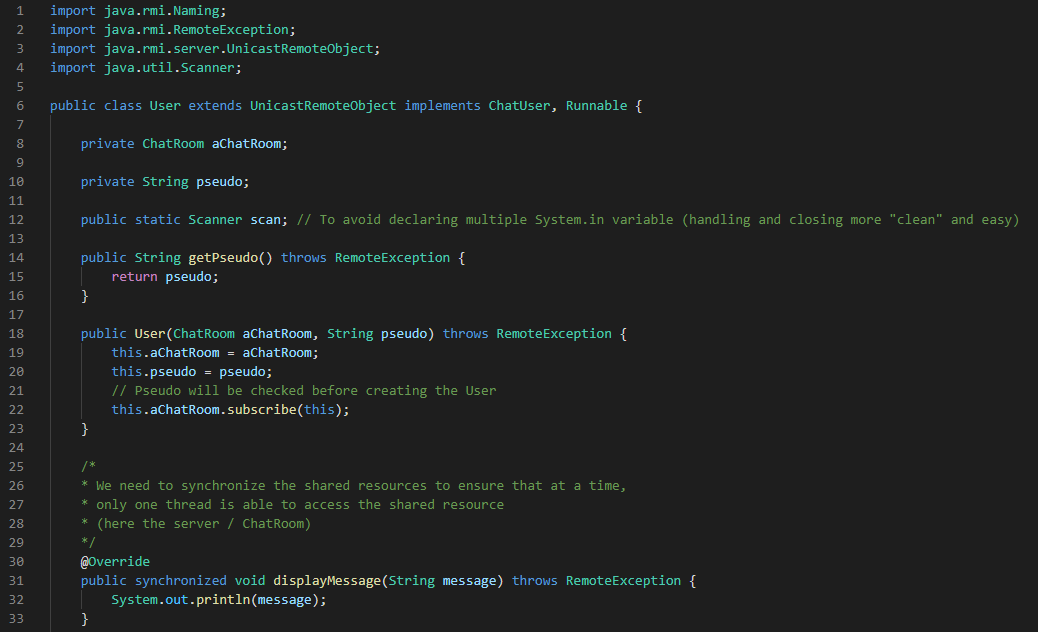


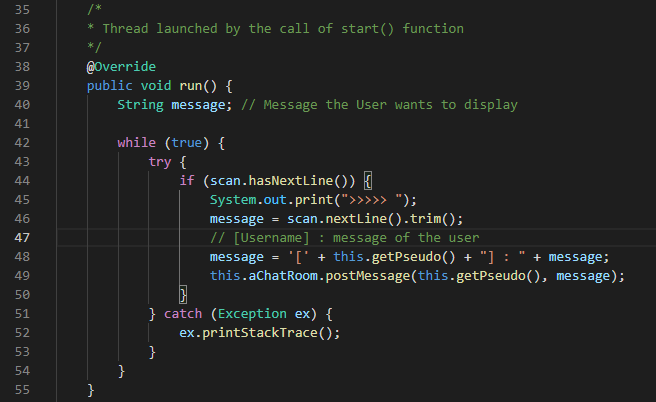
Vient ensuite le fichier ***ChatUser.java*** avec l’interface ***ChatUser*** contenant les signatures de méthodes suivantes :

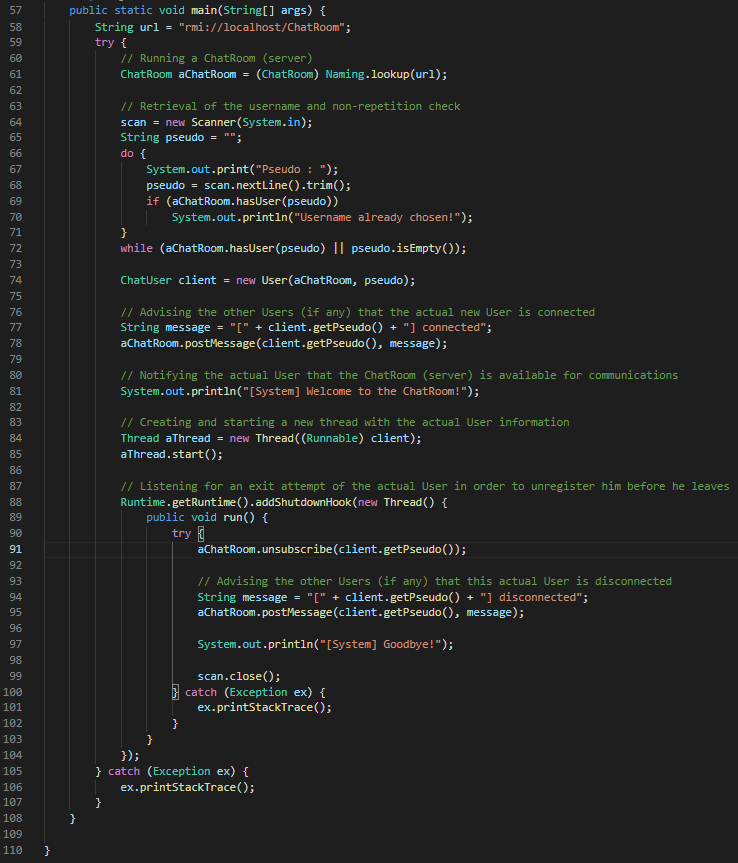


* ***getPseudo*** : retournant une chaîne de caractères contenant le pseudo de l’utilisateur appelant.
* ***displayMessage*** : recevant en paramètre un message transmis d’un autre utilisateur par l’intermédiaire du serveur, afin de l’afficher à l’écran.

Elles sont définies dans le fichier ***User.java*** contenant la classe ***User*** implémentant ***ChatUser*** et ***Runnable***. Ce dernier est dans le but de pouvoir définir la méthode ***public void run()*** qui permettra de décrire les actions d’un thread (lecture et diffusion du message de l’utilisateur dans notre cas) qui sera lancé par la méthode ***start()*** au niveau de notre ***main()***.







Le processus de lancement se fera de la façon suivante :

* Compilation des fichiers ***\*.java*** avec la commande :

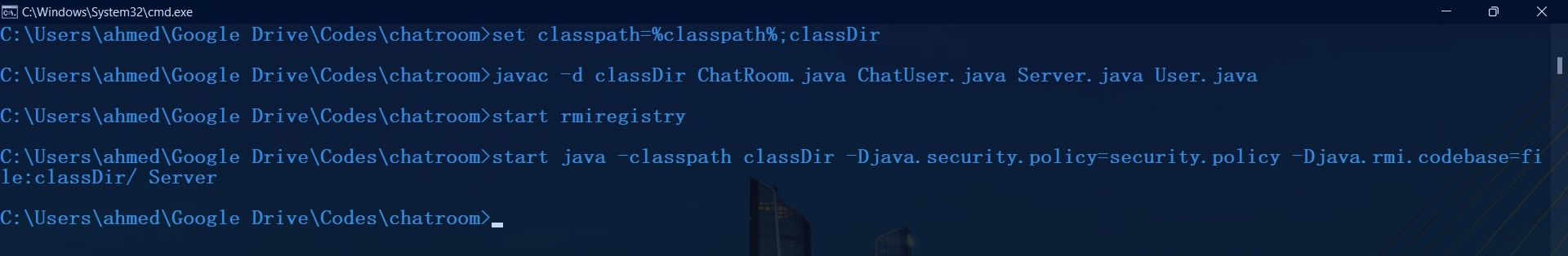
***javac -d classDir ChatUser.java ChatRoom.java Server.java User.java***

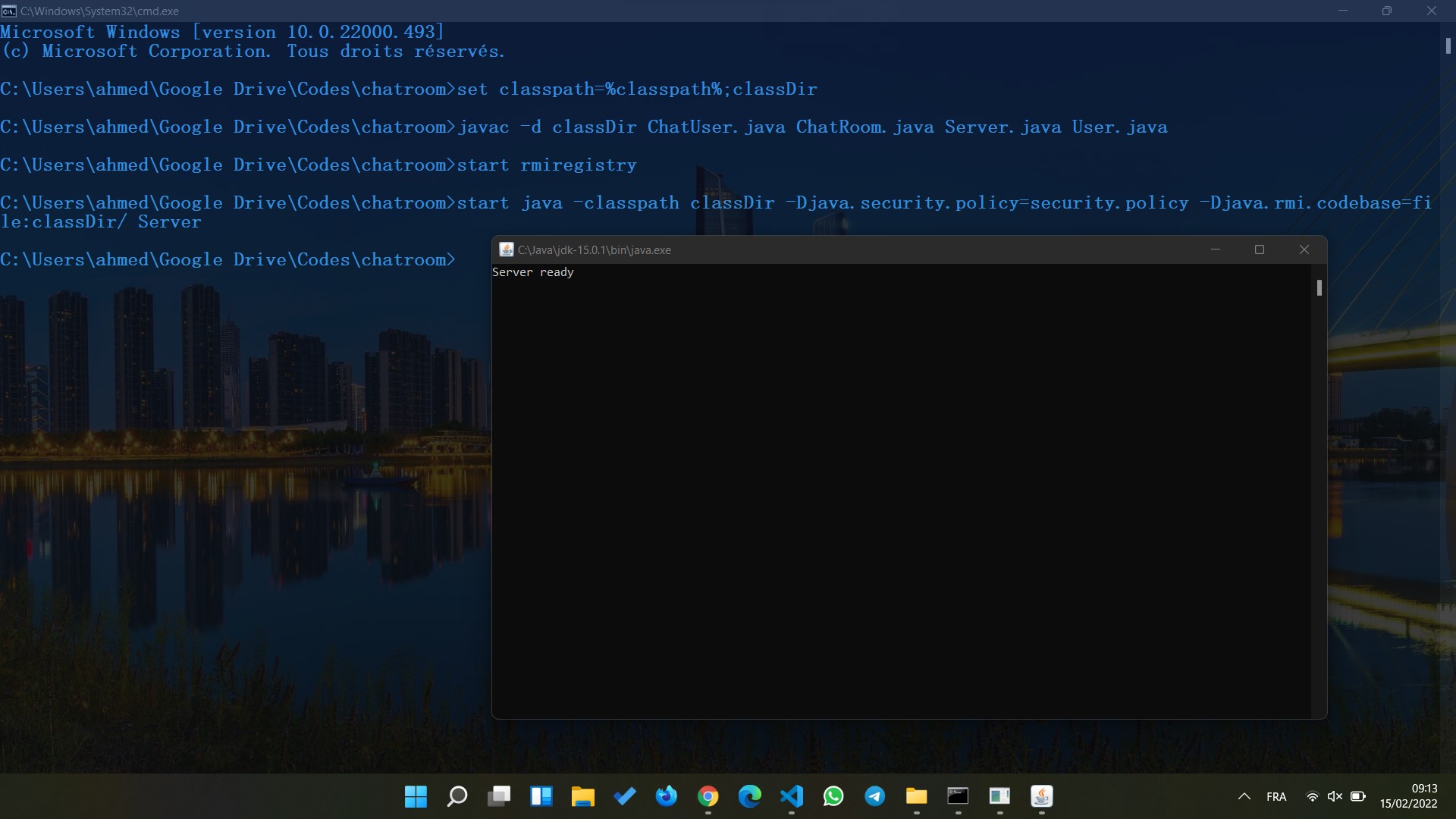
* Lancement du registre RMI :

***start rmiregistry***

* Lancement du serveur (Server) en n’oubliant pas d’inclure le fichier d’autorisation security.policy :

***java -classpath classDir -Djava.security.policy=security.policy -Djava.rmi.codebase=file:classDir/ Server***





* Lancement des clients (dans notre cas, nous avons 4 utilisateurs dans le chatroom) :

