TD4 – Java RMI

Exercice 1 – Calculator

L'objectif de l'exercice est de déployer une application distribuée programmée avec Java RMI, la calculatrice distribuée vue en cours.

1. Téléchargez le fichier Calculator.zip (disponible à l'adresse http://www.lami.univ-evry.fr/~hutzler/Cours/SyDi.htm).
2. Dézippez l'archive :
   `$ unzip Calculator.zip`
3. Compilez tous les fichiers java :
   `$ javac *.java`
4. Générez le stub et le skeleton :
   `$ rmic CalculatorImpl –keep`

   l'option `–keep` permet de conserver les fichiers java intermédiaires générés
5. Lancez le registre RMI :
   `$ rmiregistry &`
6. Démarrez le serveur :
   `$ java CalculatorServer`
7. Dans une autre console, lancez le client :
   `$ java –Djava.security.policy client.policy CalculatorClient`

   l'option `–Djava.security.policy client.policy` keep permet de définir les droits du client, à partir du fichier `client.policy`. En l'occurrence, le client doit pouvoir se connecter sur le port du registre, d'où la permission sur les sockets. Editez le fichier `client.policy` pour voir la manière dont sont définis ces droits
8. Editez les fichiers `CalculatorImpl_Stub.java` et `CalculatorImpl_Skel.java` pour voir comment les requêtes sont routées par le système de stubs et de skeletons

Exercice 2 – Bibliothèque

L'objectif de l'exercice est de concevoir une application distribuée de gestion bibliographique. Le serveur maintient à jour une base de données dans laquelle sont référencés des livres, caractérisés par leur titre (une chaîne de caractères) et un numéro ISBN (numéro identifiant unique au niveau mondial pour tous les livres édités, représenté par un entier long). Cette base de données sera représentée sous la forme d'un objet de type `Hashtable` (consulter l'API Java). Les opérations autorisées sur cette base de données sont

a. l'ajout d'un nouveau livre caractérisé par son numéro ISBN et son titre
b. la suppression d'un livre, référencé grâce à son numéro ISBN
c. l'affichage de tous les livres de la base de données

1. Quelles classes sont nécessaires ? Lesquelles doivent être distribuées ? Lesquelles peuvent être locales ?
2. Implantez l'interface `Bibliotheque.java` qui définit les opérations de gestion de la bibliothèque

3. Implantez la classe `Livre.java` et la classe `BibliothequeImpl.java`
4. Implantez le serveur `BiblioServer.java`
5. Implantez le client `BiblioClient.java` : ce dernier sera appelé avec un argument définissant le type d'opération à effectuer ("add", "remove" ou "list"). S'il s'agit de l'opération d'ajout, 2 paramètres supplémentaires sont nécessaires, le titre et le numéro ISBN ; s'il s'agit de l'opération de suppression, 1 paramètre supplémentaire est nécessaire, le numéro ISBN du livre à supprimer ; s'il s'agit de l'opération d'affichage de la liste des livres enregistrés dans la base, aucun paramètre supplémentaire n'est nécessaire.

   Exemples d'invocation :

```
$ java BiblioClient add 3712 "Java pour les nuls"
Ajout du livre [ISBN : 3712 / Titre : "Java pour les nuls"]
$ java BiblioClient add 4032 "Les systèmes distribués"
Ajout du livre [ISBN : 4032 / Titre : " Les systèmes distribués"]
$ java BiblioClient add 1234 "Au coeur de Java RMI"
Ajout du livre [ISBN : 1234 / Titre : "Au coeur de Java RMI"]
$ java BiblioClient list
[ISBN : 3712 / Titre : "Java pour les nuls"]
[ISBN : 4032 / Titre : " Les systèmes distribués"]
[ISBN : 1234 / Titre : "Au coeur de Java RMI"]
$ java BiblioClient remove 3712
Suppression du livre ISBN 3712
$ java BiblioClient list
[ISBN : 4032 / Titre : " Les systèmes distribués"]
[ISBN : 1234 / Titre : "Au coeur de Java RMI"]
```

Exercice 3 – Chat

Adaptez le principe du chat vu au TD3, et le principe de la bibliothèque pour réaliser une application de chat en java RMI