

Université d'Evry Val d'Essonne, Master Informatique et Systèmes

Conception et Programmation d'Applications Réparties

M1/CPAR

TD 3 : Objets distants (RMI)

Tarek Melliti & Pascal Poizat

2011-2012

Exercice - 1 *Prise en main : serveur de calcul*

L'objectif de l'exercice est de déployer une application distribuée programmée avec Java RMI, la calculatrice vue en cours. Afin de simuler ce qui se passerait dans un cas réel, vous commencerez par placer les classes Java dans trois répertoires différents (**client**, **serveur** et **commun**). Dans un second temps, vous pourrez utiliser plusieurs machines (à faire une fois que cela fonctionne sur une unique machine).

1. Placer l'interface **Calculator** dans le répertoire **commun**
2. Placez les classes **CalculatorImpl** et **CalculatorServer** dans le répertoire **serveur**
3. Placez la classe **CalculatorClient** dans le répertoire **client**
4. Compilez vos classes
5. Générez le *stub* à partir de la classe **CalculatorImpl** et placez le dans le répertoire **commun**
6. Regardez le contenu du fichier **CalculatorImpl_Stub.java** pour voir comment les requêtes sont routées par le système de *stubs* (attention, demander à conserver les sources dans l'étape précédente)
7. Lancez le registre RMI
8. Créez un fichier **Autho_Connexion.policies** dans lequel vous donnez tous les droits d'accès et lancez le serveur, **CalculatorServer**, en pointant l'URL du *codebase* vers le répertoire **commun**
9. Dans une autre console, lancez le client, **CalculatorServer**, avec le même fichier de droits d'accès que le serveur et en précisant également la même URL pour le *codebase*

Exercice - 2 *Réalisation d'un système de vente aux enchères*

Nous voulons réaliser un système d'enchères anglaises. Le principe de ces enchères est le suivant.

Phase d'enregistrement. L'enchère commence par l'ouverture d'une session d'enregistrement où des participants viennent s'enregistrer. L'enregistrement affecte un identifiant à chaque participant. Une fois la période d'enregistrement expirée, aucun enregistrement n'est plus possible et l'enchère peut commencer.

Phase d'enchère. L'enchère est organisée par session, chaque session possédant un prix d'ouverture. Le prix d'ouverture de la première session est le prix de départ fixé par le vendeur. Les participants, une fois une session ouverte, peuvent faire leur proposition qui doit être supérieure au prix d'ouverture de la session. Toute proposition avec un prix inférieur ou égal est ignorée. À la fermeture d'une session, le système fixe le prix d'ouverture de la prochaine session à la proposition maximale faite par les participants.

Phase d'arrêt. Le système arrête l'enchère lorsque plus aucune proposition n'est faite durant une session. Le gagnant est celui qui a fait la meilleure proposition lors de la session précédente (s'il y a lieu). Les participants sont informés de l'issue de l'enchère.

Pour réaliser ce type d'enchères, nous allons utiliser les objets distants (*RMI*) et les threads.

Les objets distants :

Pour réaliser ce système nous avons besoin de deux types d'objets distants : un pour le vendeur et un pour les participants. Leurs interfaces sont données dans la suite.

```
import java.rmi.*;
public interface Enchere extends Remote {
    public String enregistrement(Participant p) throws RemoteException ;
    public void faireProposition (String pid, float proposition) throws RemoteException ; }
```

Il s'agit de l'interface de l'objet distant qui gère le produit à vendre.

- **enregistrement** permet à un client de s'enregistrer en envoyant l'objet distant le représentant. En retour le client recevra un identifiant.
- **faireProposition** permet à un participant de faire une proposition en précisant son identifiant et la valeur de la proposition.

L'interface de l'objet distant qui représente un participant est la suivante.

```
import java.rmi.*;
public interface Participant extends Remote {
    public void debutSession(float prix, int sid) throws RemoteException ;
    public void finSession(int sid) throws RemoteException ;
    public void finEnchereResultat(boolean estGagnant) throws RemoteException ; }
```

Ces trois méthodes servent essentiellement au serveur d'enchères à informer les participants lors des différentes phases de l'enchère.

- **debutSession** informe un participant que la session numéro *sid* est ouverte avec comme prix de base *a*.
- **finSession** informe un participant que la session *sid* est terminée.
- **finEnchereResultat** informe un participant de la fin de l'enchère avec le résultat concernant le participant comme paramètre.

Les processus :

Du côté acheteur comme vendeur, chaque objet distant est géré par un processus. Les processus ont pour rôle de gérer le déroulement des différentes étapes de l'enchère.

Le processus Vendeur : "**EnchereServer**" est un processus qui gère l'ensemble de l'enchère. Chaque produit est confié à un processus "**EnchereServer**". Le comportement de ce processus est comme suit : premièrement il construit l'objet distant associé au produit et le déploie sur le registre. Ensuite il s'endort durant la période d'enregistrement. Une fois la période expiré, il se réveille et change l'état interne de l'objet distant pour ne plus accepter d'enregistrements. De manière répétitive, il ouvre une session et informe l'ensemble des acheteurs (à travers les objets distants participants) et puis il s'endort durant la période de la session. À la fin de la session il parse l'ensemble des propositions, calcule le prix de la session suivante et informe l'ensemble des acheteurs.

Le processus Acheteur : "**EnchereClient**" est le processus qui gère le comportement du client. Il maintient une référence vers l'adresse de l'objet distant du vendeur. Le processus commence par obtenir l'objet distant correspondant à l'article, ensuite il crée un objet de type participant et tente un enregistrement. Si l'enregistrement réussit alors le processus se met en attente de l'ouverture d'une session. Le processus donne alors la main à l'utilisateur de saisir une proposition. Le mécanisme est répété tant que l'enchère ne s'est pas terminée. Une fois terminé, le processus affiche le résultat.

Le code incomplet de ce programme se trouve sous le répertoire (package) *encherermi* à l'adresse <http://www.ibisc.fr/~tmelliti/cours/CPAR/TD3CPAR/>.

- 1- Complétez l'implémentation des interfaces "**Enchere**" et "**Participant**" c.à.d "**EnchereImpl**" et "**ParticipantImpl**".
- 2- Donnez le diagramme de séquence des échanges entre un acheteur ("**EnchereClient**") et le vendeur ("**EnchereServer**").
- 3- Écrivez le code des deux processus "**EnchereServer**" et "**EnchereClient**".

4- Testez votre application