

Travaux Pratiques .4. Java RMI

(Fabrique)

Lien utile:

<https://docs.oracle.com/javase/8/docs/technotes/guides/rmi/Factory.html>

Fabrique d'objets (Factory):

Rappel: Permettre au client de construire des instances multiples d'une classe C Sur le site serveur.

1. On vise mettre en œuvre une fabrique d'inversion de chaînes de caractères (Reverse1) et créer l'annuaire « rmiregistry » depuis le serveur d'objets lui-même pour éviter qu'on le lance manuellement à partir du terminal ou du cmd (interpréteur de commandes windows).

Rappel:

- Définir une interface distante (**AppInterface.java**).
- Créer une classe implémentant cette interface (**AppImpl.java**).
- Définir l'interface de fabrique d'objets (**FabInterface.java**).
- Créer une classe implémentant la fabrique d'objets (**la classe Fabrique FabImpl.java**).
- Compiler.
- Créer une application serveur (**AppServer.java**).
- Compiler l'application serveur.
- Créer les classes stub et skeleton à l'aide de **rmic**:
AppImpl_Stub.java et **AppImpl_Skel.java**
FabImpl_Stub.java et **FabImpl_Skel.java**
- Lancer le serveur **AppServer** pour la création d'objets et leur enregistrement dans rmiregistry.
- Créer une classe cliente qui appelle des méthodes distantes de l'objet distant (**AppClient.java**).
- Copier-Coller les deux Stub et les Interfaces.class du dossier « server » dans le dossier « client ».
- Compiler **AppClient.java** et la lancer.

Reverse1 : Les classes à ajouter dans le dossier « server » sont :

```
import java.rmi.*;
public interface FabReverseInterface extends Remote{
    public ReverseInterface newReverse() throws RemoteException ;}
```

```
import java.rmi.*;
import java.rmi.server.*;
public class FabReverseImpl extends UnicastRemoteObject implements
FabReverseInterface{
    public FabReverseImpl() throws RemoteException {};
    public ReverseInterface newReverse() throws RemoteException {
        return new Reverse();}
```

E-mail: thouraya.louati@gmail.com Web: <https://sites.google.com/site/thlouati/home>

Les classes à modifier sont :

```
import java.rmi.*;
import java.rmi.server.*;
import java.rmi.registry.*;
public class ReverseServer {
    public static void main(String[] args) {
        try {
            Registry registry = LocateRegistry.createRegistry(1099);

            System.out.println( "Serveur : Construction de
l'implementation");
            FabReverseImpl fab=new FabReverseImpl();
            System.out.println("Objet Fabrique lie dans le RMIREgistry");

            registry.rebind ("Fabrique",fab);
            System.out.println ("Serveur prêt." );
            System.out.println("Attente des invocations des clients ...");
        }
        catch (Exception e) {
            System.out.println("Erreur de liaison de l'objet Fabrique");
            System.out.println(e.toString());}}
```

```
import java.rmi.*;
import java.rmi.registry.*;
public class ReverseClient {
    public static void main (String [] args) {

        if(args.length != 2)
        {
            System.out.println("Utilisation : java ReverseClient ch1 ch2");
            System.exit(0);
        }
        try{
            Registry reg = LocateRegistry.getRegistry("localhost",1099);
            FabReverseInterface fabrique = (FabReverseInterface)
            reg.lookup ("Fabrique");

            ReverseInterface ReverseObj1;
            ReverseObj1= (ReverseInterface) fabrique.newReverse();
            ReverseInterface ReverseObj2;
            ReverseObj2= (ReverseInterface) fabrique.newReverse();
            String result = ReverseObj1.reverseString(args[0]);
            System.out.println("L'inverse de "+args[0]+" est "+result+"\n");
            String result1 = ReverseObj2.reverseString(args[1]);
            System.out.println("L'inverse de "+args[1]+" est "+result1+"\n");
        }
        catch (Exception e) {
            System.out.println ("Erreur d'accès a l'objet distant.");
            System.out.println (e.toString());
        }
    }
}
```

E-mail: thouraya.louati@gmail.com Web: <https://sites.google.com/site/thlouati/home>

Les commandes nécessaires pour lancer l'application Reverse1 (avec Factory) sont :

<pre>***** /***** <u>Serveur</u> *****/ //Aller dans le repertoire server user\$ cd server //Compiler les fichiers .java user\$ javac *.java //Générer les souches client et serveur Reverse_Stub.class et Reverse_Skel.class user\$ rmic -v1.1 Reverse //Générer les souches user\$ rmic -v1.1 FabReverseImpl //Lancer le serveur d'objets user\$ java ReverseServer</pre>	<pre>***** <u>Client</u> *****/ //Aller dans le répertoire du client user\$ cd ../client //Copier l'interface ReverseInteface.class dans le répertoire du client user\$ cp ../server/ReverseInterface.class . //Copier le stub client ReverseStub.class dans le répertoire du client user\$ cp ../server/Reverse_Stub.class . //Copier l'interface FabReverseInterface.class user\$ cp ../server/FabReverseInterface.class . // Copier le stub FabReverseImpl_Stub.class user\$ cp ../server/FabReverseImpl_Stub.class . //Compiler le client ReverseClient.java user\$ javac ReverseClient.java //Lancer le client user\$ java ReverseClient BONJOUR IF4</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Lancer dans un nouveau terminal la classe RegistryViewer.java (en parallèle avec le serveur d'objets). Vous trouvez cette classe dans le dossier Reverse1.

```
Compilation: javac RegistryViewer.java
Exécution: java RegistryViewer localhost port
```

Rappel: port : est le port sur lequel est lancé rmiregistry.

- b-** Proposez une mise en œuvre d'une fabrique de comptes bancaires (voir Exercice 3 du TD). Ceux qui n'ont pas une version exécutable de l'exercice 3 (version simple) peuvent récupérer le code à partir du CD.
- c-** Proposez une mise en œuvre d'une fabrique de l'application RMI HelloApp.
- d-** Proposez une mise en œuvre d'une fabrique de Compteurs accessibles à distance (Exercice 4 du TD).