

Exercice d'application en Java RMI

1) La structure des interfaces :

Serveur :

addinterface :

```
package ser;
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface addinterface extends Remote {
    int add(int a,int b) throws RemoteException; }
```

nboccurrencesinterface :

```
package ser;
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface nboccurrencesinterface extends Remote {
    public int nboccurrences(String c, String ch) throws
    RemoteException;
}
```

Client :

addinterface :

```
package client;
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface addinterface extends Remote {
    int add(int a,int b) throws RemoteException; }
```

nboccurrencesinterface :

```
package client;
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface nboccurrencesinterface extends Remote {
    public int nboccurrences(String car, String chaine) throws
    RemoteException; }
```

2) Gestion des erreurs :

Addclass :

```
public int add(int a,int b) throws RemoteException{  
    return a+b; }
```

Nboccurencesinterface :

```
public int nboccurences(String c,String ch) throws  
RemoteException{  
    if (ch == "")  
        throw new RemoteException("chaine ne doit pas etre vide");  
    else if (c == "")  
        throw new RemoteException("Saisir un caractere");  
    int longueur=ch.length();  
    int Nb=0;  
    for (int i=0; i<longueur; i++)  
        {if ((ch.substring(i, i+1)).equals(c))  
            Nb++;}  
    return Nb; }
```

3) La structure des classes Java :

AddClass :

```
package ser;  
import java.rmi.RemoteException;  
import java.rmi.*;  
import java.rmi.server.*;  
public class Addclass extends UnicastRemoteObject implements  
addinterface {  
    public Addclass() throws RemoteException {  
        super();  
    }  
    public int add(int a,int b) throws RemoteException{  
        return a+b;  
    }  
  
}
```

Nboccurencesclass :

```
package ser;
import java.rmi.RemoteException;
import java.rmi.*;
import java.rmi.server.*;
public class Nboccurencesclass extends UnicastRemoteObject
implements nboccurencesinterface {
    public Nboccurencesclass() throws RemoteException {
        super();
    }
    public int nboccurences(String c,String ch) throws
RemoteException{
        if (ch == "")
            throw new RemoteException("chaine ne doit pas
etre vide");
        else if (c == "")
            throw new RemoteException("Saisir un
caractere");
        int longueur=ch.length();
        int Nb=0;
        for (int i=0; i<longueur; i++)
        {
            if ((ch.substring(i, i+1)).equals(c))
                Nb++;
        }
        return Nb;
    }
}
```

4) La classe du serveur :

ServeurClass :

```
package ser;
import java.rmi.*;
import java.rmi.server.*;
public class Serveurclass {

    public static void main(String[] args) {
        try {
            System.out.println("Serveur : Construction de
l'implémentation");
        }
    }
}
```

```

        Addclass add= new Addclass();
        System.out.println("Objet Add lié dans RMIregistry");

Naming.rebind("rmi://clementine.cnam.fr:2001/Addition", add);
        Nboccurencesclass nboccurences = new
Nboccurencesclass();
        System.out.println("Objet Nboccurences lié dans
RMIregistry");

Naming.rebind("rmi://clementine.cnam.fr:2001/Chaine",nboccurences );
        System.out.println("Attente des invocations des clients
");
    }
    catch (Exception e) {
        System.out.println("Erreur de liaison de l'objet Objet
Nboccurences ou Objet Add");
        System.out.println(e.toString());
    }
}
}

```

5) Les commandes à lancer sur la machine clementine :

Compiler les sources (interface, implémentation de l'objet et le serveur) :

clementine> javac *.java

Lancer rmic sur Addclass :

clementine>rmic -v1.2 Addclass

Lancer rmic sur Nboccurencesclass :

clementine>rmic -v1.2 Nboccurencesclass

Démarrer rmiregistry :

clementine>rmiregistry -J-Djava.security.policy=client1.policy&

Lancer le serveur :

clementine>java ReverseServer &

6) Le programme du client :

```
package client;
import client.nboccurrencesinterface;
import client.addinterface;
import java.io.*;
import java.rmi.*;
import java.util.Scanner;
public class Client {
    public static void main(String[] args) throws IOException {
        try{
            int nb=0;
            Scanner ch = new Scanner(System.in);
            System.out.println("Saisir une chaine ");
            String chaine = ch.nextLine();
            Scanner c = new Scanner(System.in);
            System.out.println("Saisir un caractere ");
            String car = c.nextLine();
            nboccurrencesinterface nboccu =
(nboccurrencesinterface)Naming.lookup("rmi://clementine.cnam.fr:2001/
Chaine");
            nb = nboccu.nboccurences(car, chaine);
            System.out.println(" Dans la phrase "+ch+", il y a "+nb+"
occurrences de " + car);
        }
        catch (Exception e)
        {System.out.println ("Erreur d'accès à l'objet distant.");
        System.out.println (e.toString());
        }

        try{
            Scanner sc1 = new Scanner(System.in);
            System.out.println("Saisir une nombre ");
            int a = sc1.nextInt();
            Scanner sc2 = new Scanner(System.in);
            System.out.println("Saisir un autre nombre ");
            int b = sc2.nextInt();
            int somme =0;
            addinterface Add= (addinterface)
Naming.lookup("rmi://clementine.cnam.fr:2001/Addition");
            somme = Add.add(a,b);
            System.out.println(" la somme de "+a+" + "+b+" = " +
somme);
```

```
    }  
    catch (Exception e)  
    {System.out.println ("Erreur d'accès à l'objet distant.");  
      System.out.println (e.toString());}  
    }  
}
```

Exécution du client :

client>java -Djava.security.policy=client1.policy Client