Introduction à RMI

Jean-Marc Farinone
`farinone@cnam.fr`

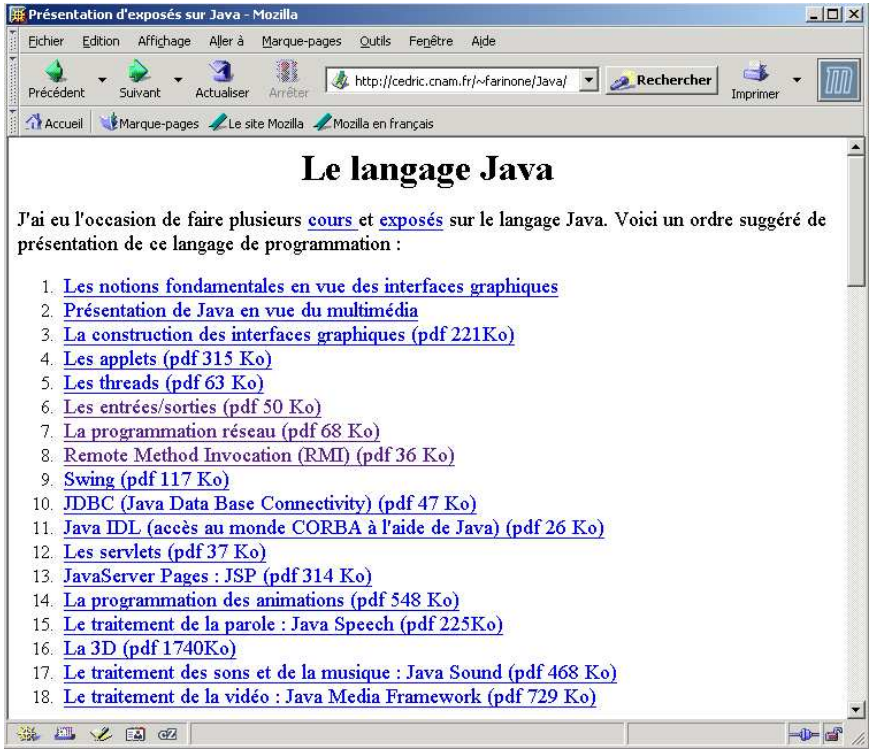
Maître de Conférences
Conservatoire National des Arts et Métiers
CNAM Paris (France)

Plan de l'exposé

- Rappel programmation réseau
- Rappel programmation écriture/lecture d'objets
- Réseau + écriture/lecture d'objets : une synthèse
- Conclusion : appel de méthodes à distance

Rappel

- En Java on peut faire facilement de la programmation :
  - réseau (si, si !) TCP ou UDP ou multicast
  - écrire/lire des objets dans tout endroit où on peut écrire/lire



- Voir à `http://cedric.cnam.fr/~farinone/Java`,
  - cours (en pdf) programmation réseau,
  - cours (en pdf) entrées sorties pour écrire/lire des objets

Plus précisément : programmation réseau (1/2)

- Pour un serveur TCP, on a la structure de code :

```
ServerSocket listen_socket = new ServerSocket(port);
Socket client = listen_socket.accept();
BufferedReader in = new BufferedReader (new
     InputStreamReader(client.getInputStream()));
PrintStream out = new
     PrintStream(client.getOutputStream());
// puis des instructions comme
String uneRequete = in.readLine();
// traitement de la requete et fabrication de uneReponse
out.println(uneReponse);
```

- Pour un client TCP, on a la structure de code :

```
Socket socket = new Socket(machineDist, port);
BufferedReader in =
     new BufferedReader (new InputStreamReader(socket.getInputStream()));
PrintStream out = new PrintStream(socket.getOutputStream());
// puis des instructions comme
out.println(uneRequete);
String laReponse = in.readLine();
```

Plus précisément : programmation réseau (2/2)

- En fait les serveurs sont multithreadés et on a plutôt la structure de code :

```
le serveur écoute

lorsqu'un client se connecte, une thread est créée pour
traiter sa requête

retour de l'écoute.
```

Plus précisément : écriture/lecture d'objets (1/2)

- En Java, les données (attributs, propriétés) des objets peuvent être écrites et relues (donc par exemple sauvegardées) : pas les méthodes ou constructeurs.
- On parle de sérialisation
- Si un objet contient une référence sur un objet, cet objet inclus est aussi sérialisé et on obtient ainsi un graphe de sérialisation d'objets.
- Pour indiquer que les objets d'une classe peuvent être persistents on indique que cette classe implémente l'interface `Serializable`. Il n'y a aucune méthode à implanter dans cet interface qui joue seulement le rôle de marqueur pour dire que les objets peuvent être sérialisés.

Plus précisément : écriture/lecture d'objets (2/2)

- Ecrire un objet :

```
import java.util.*;
import java.io.*;
public class EcritObjetPers {
    public static void main(String args[]) {
     Date d = new Date();
     FileOutputStream f;
     ObjectOutputStream s;
     try {
               f = new FileOutputStream ("date.ser");
               s = new ObjectOutputStream(f);
               s.writeObject(d);
               s.close();
     } catch (IOException e) { }
    }
}
```

- Lire un objet :

```
import java.util.*;
import java.io.*;
public class LitObjetPers {
    public static void main(String args[]) {
         Date d = null;
         FileInputStream f;
         ObjectInputStream s;
         try {
               f = new FileInputStream ("date.ser");
               s = new ObjectInputStream(f);
               d = (Date) s.readObject();
               s.close();
         } catch (Exception e) { }
    }
}
```

Réseau + écriture/lecture d'objets : une synthèse

- Si au lieu d'écrire dans un fichier `.ser`, on écrit dans un canal de communication entre deux machines.
- Si au lieu de lire dans un fichier `.ser`, on lit depuis ce même canal de communication.
- On va pouvoir transférer des objets sur le réseau …

- … entre deux programmes Java ...
- … tournant sur deux machines distinctes et (éventuellement très !!) distantes ...
- … ayant des systèmes d'exploitation, des architectures éventuellement (très !!) différentes.
- (mais ayant des machines virtuelles Java)

Exécuter des méthodes à distance

- Pour exécuter des méthodes à distance de la forme :

```
…méthode(refObjet1, refObjet2, varDeTypePrimitif, …) ;
```

- , il suffit de définir une bonne fois pour toute un protocole de lancement des méthodes (JRMP : Java Remote Method Protocol). Par exemple, d'abord le nom de la méthode puis passer les arguments en les copiant dans le canal

Bibliographie RMI

- Site originel
  `http://java.sun.com/products/jdk/rmi/index.jsp`

Fin