

Année : 2010-2011

Formation : SI4

Examen : App. Réparties

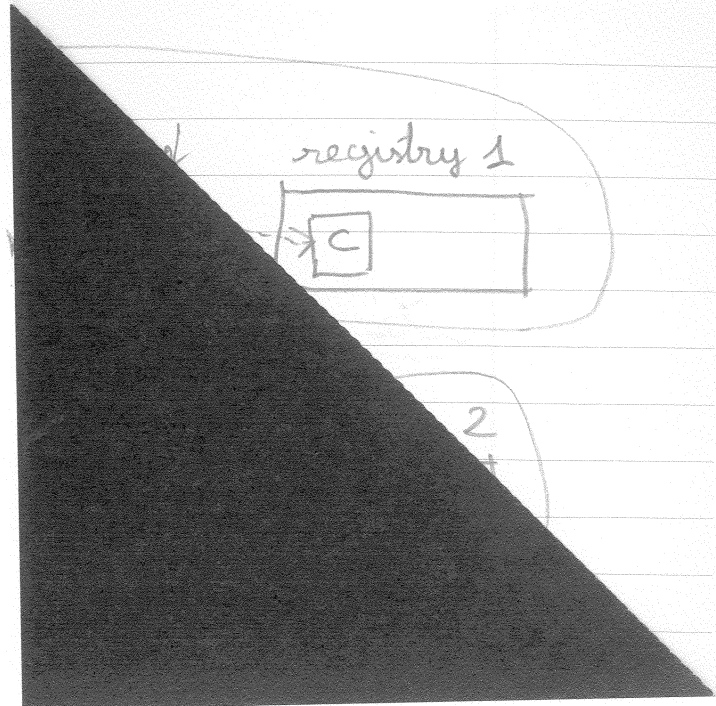
N° étudiant : 20905400

Note de l'épreuve

48

(1) Le Candidat doit inscrire ici : ses nom, prénoms, lieu et date de naissance, puis rabattre suivant le pointillé le coin de la copie et coller.

Il est interdit au Candidat de signer sa copie ou d'y inscrire un signe quelconque pouvant en indiquer la provenance.



Sophia , le 25/05/11

Exercice 1 :

- 1) Faux. Il est seulement possible de récupérer une référence vers celui-ci.
- 2) Vrai : Grâce au retour d'une méthode distante.
- 3) Faux. Il est seulement possible de contrôler, à l'aide du mot-clé synchronise et que deux méthodes ne soient pas exécutées simultanément.
- 4) Vrai : les types primitifs sont sérialisables. Final ou pas c'est indifférent. Pour passer à type non primitif il faut les déclarer sérialisables.
- 5) Faux. On ne rend accessibles que les méthodes contenues dans l'interface distante (qui étend Remote).
- 6) Vrai : grâce à RMI-IIOP.

Exercice 2 :

- 1) Faux. Ce n'est pas JAFS qui permet de récupérer les objets. Il s'agit plutôt de définir des "factory" SSL à la création d'un objet UnicastRemote Object.
- 2) Vrai : Le module de Login est celui qui vérifie le mot de passe.
- 3) Faux. La méthode login est effectivement appelée. En revanche, la méthode commit est pas explicitement appelée.

5,5 / 6

1

1

1

0,5

1

1

4,5 / 6

1

1

1

Côté

0

4) Vrai Si, il faut avoir le droit d'ajouter 1 principal, d'exécuter du code en privilégé (si besoin), mais surtout la permission `security.auth`

0,5

5) Vrai : les types sont écrits dans le langage IDL. <sup>login, config = "nom fichier de login"</sup>  
Lors du mapping, ils seront traduits automatiquement. Mais, il est quand même souhaitable de connaître le mapping, car il y a certaines implications sur les types utilisés dans son appli

1

6) Faux : L'ensemble des objets CORBA sont manipulés de la même façon, avec les classes du package `org.omg.CORBA`.

### Exercice 3 :

1) Il y a un seul téléchargement de classe :

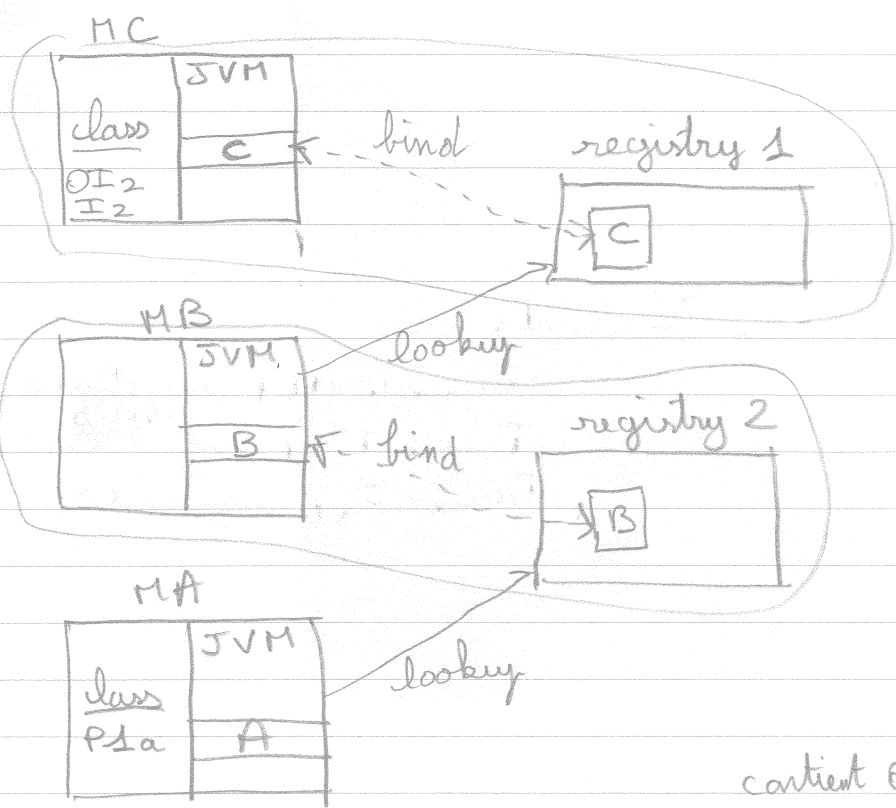
- A connaît l'interface de B, donc  $I_1$  et  $P_1$
  - B ne connaît pas  $P_1$  a.
- ↳ il va le télécharger auprès de A.

2) Si encore C ne connaît pas  $P_1$  a, il va la télécharger auprès de A.

3) Le serveur B va télécharger  $OI_2$  et donc  $I_2$  auprès de C.

4) Le client A va télécharger  $OI_2$  et donc  $I_2$  auprès de C.

5)



h/h

contient OI2.class  
I2.class

6) Machine C: \$ rmiregistry 2001 &  
 \$ java ClassServer 2002 binclient/  
 \$ java -Dserver.codebase=http://...:2002/

Machine B: \$ rmiregistry 2005 &

Machine A: \$ java ClassServer 2009 binclient/  
 \$ java -Dserver.codebase=http://...:2009/

↑  
contient  
Psa.class

Exercise 4:

Côté serveur:

- Trace: B created
- Trace A: created
- Trace: Broker binded
- Trace: B () say Hello: unknown

h/h

Côté client:

Trace: A() say Hello: SERVEUR

Trace: E() say Hello: SERVEUR

Trace: J() say Hello: unknown.