Université des sciences et technologies de Lille (USTL) Janvier 2008
Master mention Informatique – M1
Module : Conception d'applications réparties (CAR)

# Examen

# Conception d'applications réparties (CAR)

Tous documents papier autorisés. Téléphones et ordinateurs interdits.
Le barème est donné à titre indicatif. 3 heures.
Ce sujet comporte 3 pages.

## 1 Questions de cours (3 points)

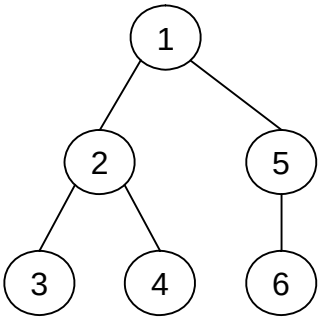
1.1 En Java RMI, que permet de faire la méthode `Naming.lookup` ? Quelle interface est implémentée par l'objet retourné par `Naming.lookup` ? Quelle est la classe de l'objet retourné par `Naming.lookup` ?

1.2 La transmission d'objets en Java RMI se fait-elle par référence (si oui, dans quel cas) ? par copie (si oui, dans quel cas) ?

1.3 Lorsque l'on souhaite transmettre un graphe d'objets en Java RMI, à quoi sert l'interface `Serializable` ?

## 2 Diffusion de messages (11 points)

On considère un protocole client/serveur qui permet de diffuser des messages à un ensemble de sites organisés selon une topologie en arbre. Chaque nœud de l'arbre propage le message à ses fils. Par exemple, dans la figure ci-dessous, pour diffuser son message, le site 1 l'envoie en parallèle à 2 et 5, puis 2 l'envoie en parallèle à 3 et 4 tandis que 5 l'envoie à 6.




2.1 Quel peut-être l'avantage d'une diffusion en arbre par rapport à une diffusion habituelle (par exemple de type Multicast IP) ? (1 point)

On considère que les sites communiquent en UDP sur le port 5000 en envoyant des messages de 2048 octets (contenu quelconque), et que chaque site a :

- une variable `pere` qui contient l'adresse de son père dans l'arbre (`null` si pas de père),
- un tableau `fils` qui contient les adresses de ses fils (tableau vide si pas de fils).

2.2 Ecrire en Java le code d'un nœud intermédiaire (dans l'exemple 2 et 5 sont des nœuds intermédiaires). (2 points)

On suppose maintenant que les nœuds feuille (3, 4 et 6 dans l'exemple) renvoient à l'émetteur un accusé de réception (message de 8 octets contenant des données quelconque) lorsque le message arrive. Lorsqu'un nœud intermédiaire a reçu les accusés de notification de tous ses fils, il renvoie à son propre père (1 dans l'exemple) un accusé de réception (message de 8 octets contenant des données quelconque).

2.3 Compléter le code Java de la question précédente en y incluant ce comportement. (1 point)

## RMI

On considère maintenant que les nœuds de l'arbre ne communiquent plus par UDP mais via RMI. On considère également que les diffusions ne se font plus uniquement à l'initiative du site racine (1), mais que n'importe quel site de l'arbre peut initier une diffusion. Chaque site a une variable `pere` qui est maintenant une référence d'objet RMI et un tableau `fils` qui est un tableau d'objets RMI. Les messages diffusés sont des tableaux d'octets. Les accusés de réception sont des booléens.

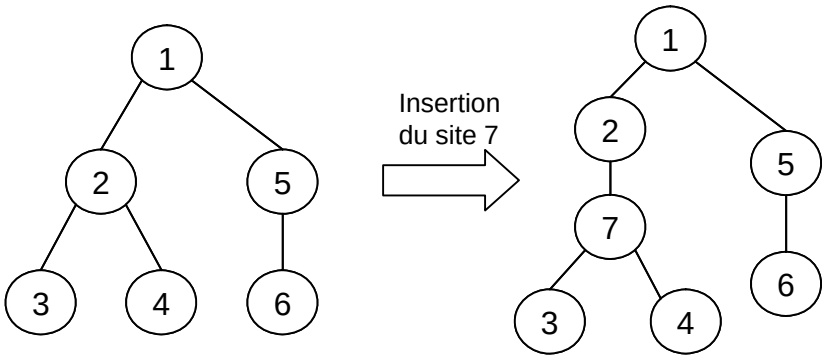
2.4 Proposer en français un algorithme qui permet de diffuser un message à tous les sites de l'arbre à partir de n'importe quel site (on rappelle qu'aucun site ne connaît la topologie complète de l'arbre, mais que chaque site ne connaît que son père et ses fils). (1 point)

2.5 Proposer une interface RMI (`SiteItf`) et sa classe d'implémentation (`SiteImpl`) mettant en œuvre cet algorithmique. L'interface et la classe doivent pouvoir servir pour tous les sites de l'arbre. (2 points)

2.6 Votre solution permet-elle de supporter plusieurs diffusions initiées simultanément par différents sites de l'arbre. Si oui pourquoi ? Si non, quelle modification faudrait-il apporter à la solution ? (1 point)

## *Servlet*

On souhaite mettre en place une servlet pour gérer l'arbre de diffusion. La servlet doit permettre d'insérer et de retirer un site dans l'arbre. La servlet interagit avec les sites de l'arbre via Java RMI. À titre d'exemple, on souhaite que la servlet puisse insérer un site 7 comme illustré ci-dessous.




2.7 Que faut-il modifier au niveau de l'interface `SiteItf` et de la classe `SiteImpl` pour permettre à la servlet de fonctionner ? (0,5 point)

2.8 On suppose que l'on dispose d'un formulaire permettant de saisir le nom de l'objet RMI (paramètre `nouveauSite`) que l'on veut insérer dans l'arbre et le nom de l'objet père (paramètre `sitePere`) auquel on veut attacher cet objet. Écrire le code Java de la servlet qui réalise l'insertion. (1,5 point)

2.9 Que se passe-t-il en cas de modification concurrente par différentes personnes de la structure de l'arbre ? Que faudrait-il faire pour résoudre le problème ? (1 point)

## 3 Ping-pong CORBA (4 points)

Soient deux objets CORBA A et B :
- l'objet B implémente l'interface `BItf` qui fournit une méthode `ping` prenant en paramètre d'entrée une référence d'objet CORBA de type `AItf`. La méthode `ping` ne retourne rien (type de retour `void`). La méthode `ping` appelle la méthode `pong` sur la référence passée en paramètre.
- l'objet A implémente l'interface `AItf` qui fournit une méthode `pong` (aucun paramètre, type de retour `void`). La méthode `pong` est vide.

3.1 Écrire le code CORBA IDL des interfaces `AItf` et `BItf`. (1 point)

3.2 Écrire des classes `AImpl` et `BImpl` qui implémentent respectivement, les interfaces `AItf` et `BItf`. Écrire le code de la méthode main de la classe Client qui instancie les objets A et B et appelle la méthode `ping` sur l'objet A. (1 point)

3.3 On souhaite mesurer le temps mis par les invocations de méthodes pour faire l'aller-retour (appel de `ping` suivi de `pong`). À titre indicatif, la méthode Java `System.currentTimeMillis()` retourne un entier long qui correspond à l'heure courante. Quelle méthode faut-il modifier pour réaliser cette mesure ? Donner le code de cette modification. (0,5 point)

3.4 On considère maintenant que les méthodes `ping` et `pong` sont `oneway`. Peut-on toujours mesurer le temps pour faire l'aller-retour (appel de `ping` suivi de `pong`) de la même façon que précédemment ? Justifier votre réponse. En cas de changement, proposer une modification du code. (1,5 point)

## 4 IDL CORBA (2 points)

Indiquer quelles sont les erreurs contenues dans l'interface IDL suivante. Expliquer brièvement en quoi consistent les erreurs et proposer une correction. (2 points)

```
public interface Carte {
    const long echelle;
    typedef enum {nord,sud,est,ouest} Cardinaux;
    void orientation( Cardinaux direction );
    interface Michelin {
        string numero();
        long echelle() { return echelle; }
    };
};
```