IPST-CNAM
Systèmes répartis
NFP 214
Mardi 24 Juin 2008

Sans document
Durée : **2 heures**
Enseignants : LAFORGUE Jacques

2ième Session NFP 214

CORRECTION

(COURS)

1. QCM (60 points)

Mode d'emploi :

Ce sujet est un QCM dont les questions sont de 3 natures :

- **les questions à 2 propositions**: dans ce cas une seule des 2 propositions est bonne.
  - +1 pour la réponse bonne
  - –1 pour la réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est bonne
  - + 1 pour la réponse bonne
  - –½ pour chaque réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est fausse
  - + ½ pour chaque réponse bonne
  - –1 pour la réponse fausse

Il s'agit de faire une croix dans les cases de droite en face des propositions.

On peut remarquer que cocher toutes les propositions d'une question revient à ne rien cocher du tout (égal à 0) et répondre au hasard est une très mauvaise stratégie.

Si vous devez raturer une croix, faites-le correctement afin qu'il n'y ait aucune ambiguïté.

N'oubliez pas d'inscrire en en-tête du QCM, votre nom et prénom.

Vous avez droit à 10% de points négatifs sans pénalité.

NOM:	PRENOM:

Un bus CORBA est :		Q 1
1	une marque de véhicule de transport scolaire	
2	un moyen logiciel de stockage et de transport de l'information dans une application répartie	X

Plus sérieusement.

Un système réparti est (condition nécessaire mais pas suffisante) :		Q 2
1	un système informatique dans lequel les données sont réparties à différents endroits d'un réseau informatique	X
2	un système informatique dans lequel les données sont centralisées dans une base de données et utilisées de manières distantes.	

Les moyens actuels et modernes de développement d'un système réparti utilisent les notions de la programmation objet		Q 3
1	OUI	X
2	NON	

Les composants d'un système réparti s'exécutent, généralement, d'une manière asynchrone		Q 4
1	OUI	
2	NON	X

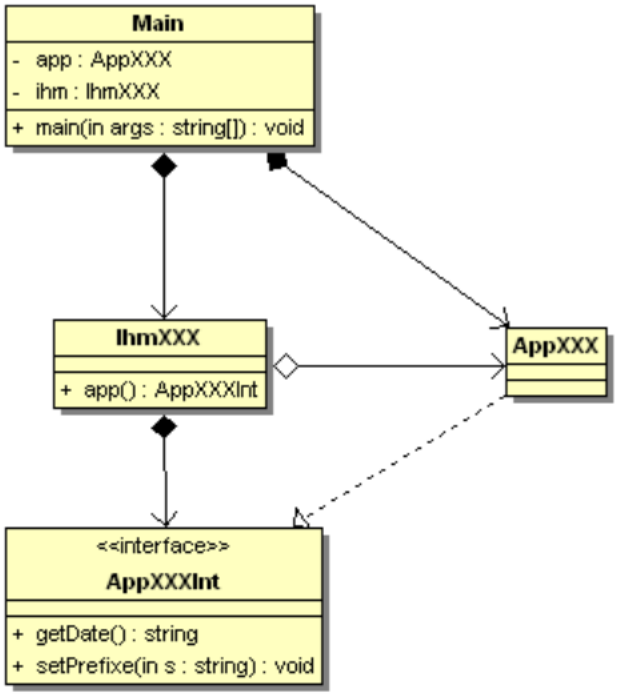
La synchronisation de l'appel d'une méthode distante est indispensable pour centraliser une donnée dans un système réparti composé de plusieurs objets distribués		Q 5
1	OUI	
2	NON	X

La protection aux intrusions extérieures d'un système réparti est bien maîtrisée par les ORB du marché		Q 6
1	OUI	
2	NON	X

Le service de nommage dans un système distribué CORBA :		Q 7
1	permet de rendre transparent, pour les clients, la localisation des objets distribués sur le réseau	X
2	sert d'intermédiaire de communication entre les clients et les serveurs	
3	centralise les échanges de données via les sockets entre tous les objets distribués du réseau	

En CORBA, le résultat de la résolution d'un IOR est un objet CORBA		Q 8
1	OUI	X
2	NON	

En CORBA, la référence d'un objet CORBA est sérialisable		Q 9
1	OUI	X
2	NON	



		Q 10
1	Ce schéma montre que l'IHM et son Applicatif associé communiquent via des méthodes distantes	
2	Ce schéma montre que l'interface AppXXXInt contient la description des méthodes que l'applicatif a le droit d'appeler	
3	Ce schéma montre que l'IHM utilise les méthodes d'un applicatif pour réaliser les traitements métiers demandés par l'utilisateur	X

Le rôle du POA dans une architecture CORBA est notamment de :		Q 11
1	centraliser le traitement des requêtes clientes	X
2	créer, modifier, supprimer un servant	X
3	délocaliser un servant d'une machine à une autre	

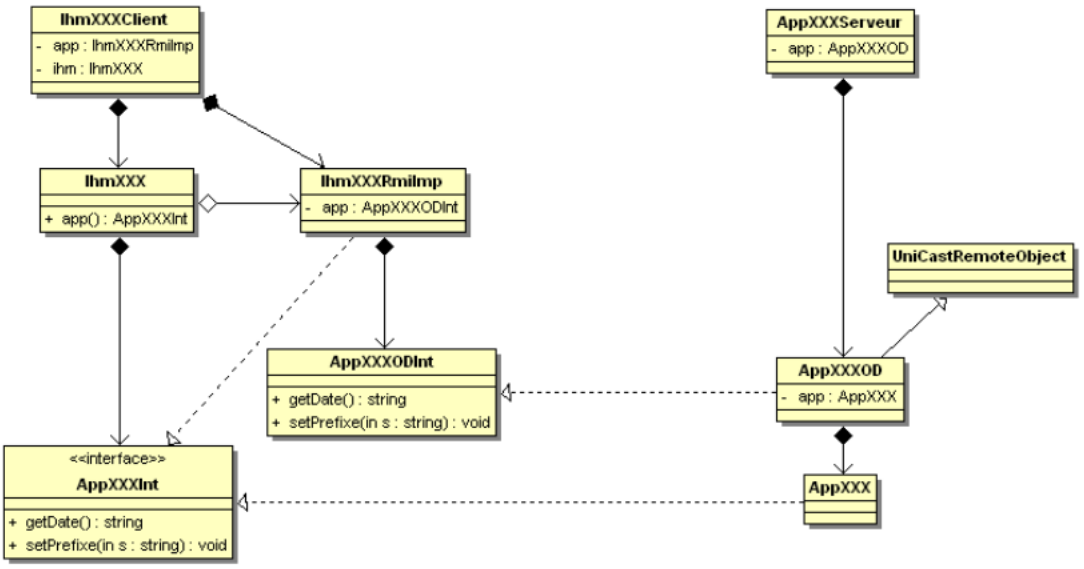
En CORBA, un servant est un objet distribué		Q 12
1	OUI	X
2	NON	

En RMI, un objet distribué		Q 13
1	est un objet qui hérite de RemoteObject	
2	est un objet qui implémente une interface de méthode distante	X
3	est un objet qui encapsule un objet qui assure la communication	

Le langage JAVA assure la sérialisation de ces données en utilisant la réflexivité du langage		Q 14
1	OUI	X
2	NON	

En CORBA, il est indispensable de passer par un service de nommage pour utiliser les méthodes d'un objet distant		Q 15
1	OUI	
2	NON	X

Le rôle de l' "interface" Java est essentiel dans la conception d'application composée d'une partie représentation (IHM) et d'une partie applicative (métier). Pourquoi ?		Q 16
1	Pour faciliter le portage de l'IHM dans différents environnements informatiques	X
2	Pour rendre indépendant le composant logiciel IHM des moyens de communication (Socket, CORBA, RMI, Http, ...)	X
3	Pour imposer une communication distante entre l'IHM et l'Applicatif	



Ce schéma vu dans le cadre de l'atelier 16 (Architectures n Tiers) décrit les dépendances des composants d'un système informatique		Q 17
1	L'applicatif utilise l'interface AppXXXInt afin de communiquer avec l'IHM	
2	L'IHM crée un objet distribué (IhmXXXClient crée IhmXXRmiImp qui à son tour crée AppXXODInt qui est un objet distribué)	
3	L'objet distribué AppXXXOD crée et encapsule l'applicatif	X

Les principes de communication distante dans un système distribué peuvent être :		Q 18
1	un client reçoit une copie de l'objet distribué distant	
2	un client reçoit une interface de l'objet distribué distant	X
3	un client reçoit un pointeur sur l'objet distribué distant	

L'adaptateur en Java RMI est un process qui s'exécute en dehors de toute JVM		Q 19
1	OUI	
2	NON	X

On peut exécuter plusieurs adaptateurs sur une même machine		Q 20
1	OUI	X
2	NON	

En CORBA, un idl est un fichier écrit dans le langage IDL spécifié par l'OMG		Q 21
1	OUI	X
2	NON	

Un IDL peut contenir la définition de plusieurs interfaces		Q 22
1	OUI	X
2	NON	

En Java RMI, l'instruction **lookup** :		Q 23
1	retourne un stub	X
2	retourne un skeleton	

En Java RMI, l'instruction **bind** :		Q 24
1	permet de créer un objet distribué	
2	permet d'enregistrer un objet distribué dans un adaptateur	X
3	alloue un port de communication pour l'objet distribué	

En CORBA, un service de nommage est utilisé par :		Q 25
1	uniquement les servants qui ont été enregistrés (bind) dans le service de nommage	
2	uniquement les servants de la machine sur lequel le service de nommage a été exécuté	
3	uniquement les servants d'un même VLAN (réseau)	X

L'IOR d'un objet CORBA est un objet Java prédéfini permettant de se connecter à un objet distribué		Q 26
1	OUI	
2	NON	X

L'IOR contient les informations de connexion (localisation réseau) d'un objet distribué		Q 27
1	OUI	X
2	NON	

Le Factory est un servant permettant de créer un objet distribué sur sollicitation distante		Q 28
1	OUI	X
2	NON	

Un bon système réparti est un système répartie qui semble centralisé mais dont ces composants ne le sont pas		Q 29
1	OUI	X
2	NON	

L'en-tête de déclaration d'une méthode **main** d'une classe JAVA peut-être:		Q 30
1	static public void main(String args[])	X
2	static public void main(int nb_args, String args[])	
3	void main(int nb_args, String args[])	

L'instruction JAVA permettant de lever une exception est:		Q 31
1	throws Exception;	
2	throw new Exception("Erreur");	X

3	catch (Exception e)
	{
	System.out.println("Erreur");
	}	

Dans une architecture distribué, un objet distribué peut-il utiliser les services d'un autre objet distribué?		Q 32
1	OUI	X
2	NON	

Choisir le(s) code(s) correct(s) de la méthode main de GrilleOD.java qui crée l'objet distribué GRILLE		Q 33
1	String hostRegistry = args[0]; String portRegistry = Integer.parseInt(args[1]); urlRegistry="rmi://"+hostRegistry+":"+portRegistry+"/"; Naming.rebind(urlRegistry+"GRILLE");	
2	String hostRegistry = args[0]; String portRegistry = Integer.parseInt(args[1]); Naming.rebind(new GrilleOD(20,20, hostRegistry, portRegistry));	
3	String hostRegistry = args[0]; String portRegistry = Integer.parseInt(args[1]); urlRegistry="rmi://"+hostRegistry+":"+portRegistry+"/"; Naming.rebind(urlRegistry+"GRILLE", new GrilleOD(20,20,urlRegistry));	X

Toutes les méthodes distantes d'un objet distribué		Q 34
1	doivent appartenir à la même interface	
2	peuvent appartenir à plusieurs interfaces	X
3	doivent toutes être implémentées dans le même objet distribué	X

Un ORB du marché, permet de créer un objet distribué à distance		Q 35
1	OUI	
2	NON	X

CORBA est un logiciel		Q 36
1	OUI	
2	NON	X

Un ORB est une implémentation de CORBA		Q 37
1	OUI	X
2	NON	

Le rôle du ORB est de		Q 38
1	sécuriser les échanges de données entre les services	
2	répartir les services et les données	X
3	rendre transparent la localisation des services	X

Les composants d'un ORB sont :		Q 39
1	un serveur, un client et une interface d'échange	
2	des APIs, un annuaire et un IDL	X

La sérialisation est utilisé dans les ORB		Q 40
1	OUI	X
2	NON	

La sérialisation permet d'écrire un objet dans :		Q 41
1	un fichier	X
2	un socket	X
3	une base de donnée	

La synchronisation des threads entre les objets distribués est indispensable dans un système réparti		Q 42
1	OUI	
2	NON	X

Une méthode distante d'un serveur qui est utilisée par plusieurs clients doit être synchronisée		Q 43
1	OUI	
2	NON	X

La synchronisation à outrance des méthodes distantes de tous les objets distribués d'un système réparti, nuit à la performance globale du système informatique		Q 44
1	OUI	X
2	NON	

La communication Web entre un client et un serveur utilise un ORB pour faire l'interface entre		Q 45
1	le client léger d'un navigateur et le serveur Http	
2	le serveur Http et les servlets	
3	les servlets et les services du système d'information	X

A cause des temps de communication entre les composants d'un système réparti, il est souvent extrêmement difficile de connaître l'état exacte des données réparties dans les différents services		Q 46
1	OUI	X
2	NON	

Des ORB différents ne peuvent pas communiquer entre eux		Q 47
1	OUI	
2	NON	X

On peut passer en paramètre d'une méthode distante, un objet distribué afin que le client puisse l'utiliser pour utiliser les méthodes distantes de cette objet distribué		Q 48
1	OUI	
2	NON	X

En Java RMI, l'instruction lookup retourne (dans le stub), entre autre, le port utilisé par l'objet distribué pour le traitement des requêtes afin que le client écrivent ses requêtes sur ce port.		Q 49
1	OUI	X
2	NON	

En Java RMI, un objet distribué est un serveur de socket dont le port est déterminé par défaut par la classe UnicastRemoteObject		Q 50
1	OUI	X
2	NON	

Le Factory est un élément qui permet de factoriser la conception des objets distribués		Q 51
1	OUI	
2	NON	X

Les avantages d'un système répartis sont :		Q 52
1	un système réparti est adapté pour donner une qualité de service identique tant local que distant	X
2	un système réparti d'une grande entreprise industrielle est plus à même d'être sécurisé qu'un système non réparti.	
3	dans un système réparti, il est envisageable de continuer un service même dégradé	X

Soit la déclaration de la classe suivante :		Q 53

	`public class MonThread extends Thread`
	`{`
	`  public void run()`
	`  {`
	`  // traitement du thread`
	`  }`
	`}`	

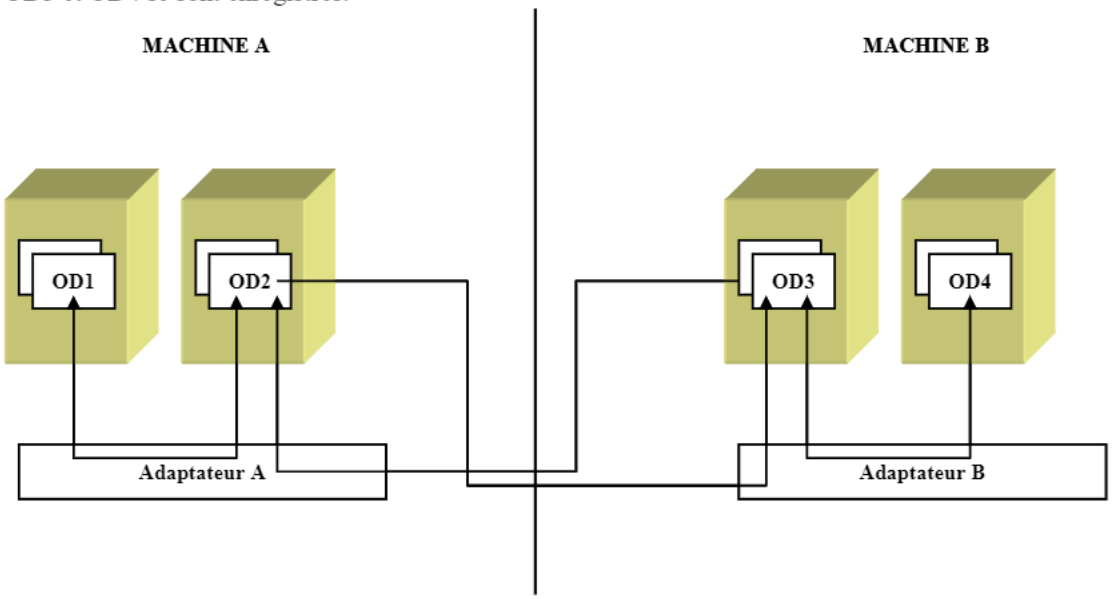
Pour créer le thread et l'exécuter, il faut faire les instructions suivantes:		
1	Runnable p = new Runnable (); MonThread q = new Thread(p); q.start();	
2	MonThread p = new MonThread (); p.start();	X
3	MonThread p = (MonThread)(new Thread()); p.start()	

Un objet passé en paramètre d'une méthode distante est reçu par l'appelant :		Q 54
1	sous la forme d'un nouvel objet qui est une copie du paramètre	X
2	sous la forme d'une référence du paramètre	

En CORBA, il est obligatoire d'enregistrer les servants (objets distribués) dans un annuaire (service de nommage)		Q 55
1	OUI	
2	NON	X

Soit, un IDL qui contient une interface de nom *InterfaceDeviseOD*. Cette interface génère une classe utilisée pour créer, par héritage, les servants. Le nom de cette classe est :		Q 56
1	InterfaceDeviseODOperations	
2	InterfaceDeviseODPOA	X
3	InterfaceDeviseODHolder	

La référence d'un objet CORBA contient l'IOR du servant associé		Q 57
1	OUI	X
2	NON	

L'architecture distribuée suivante est correcte. Le sens de la flèche correspond à l'appel d'une méthode distante (exemple: OD2-> OD3 : OD2 appelle une méthode distante de OD3). La machine A a un adaptateur dans lequel OD1 et OD2 se sont enregistrés. La machine B a un adaptateur dans le quel OD3 et OD4 se sont enregistrés.		Q 58
		
1	Le nom de l'objet OD1 doit être différent de celui de OD3	
2	Le nom de l'objet OD1 doit être différent de celui de OD2	X
3	Si l'adaptateur A est en panne, je peux enregistrer OD1 et OD2 dans l'adaptateur B.	

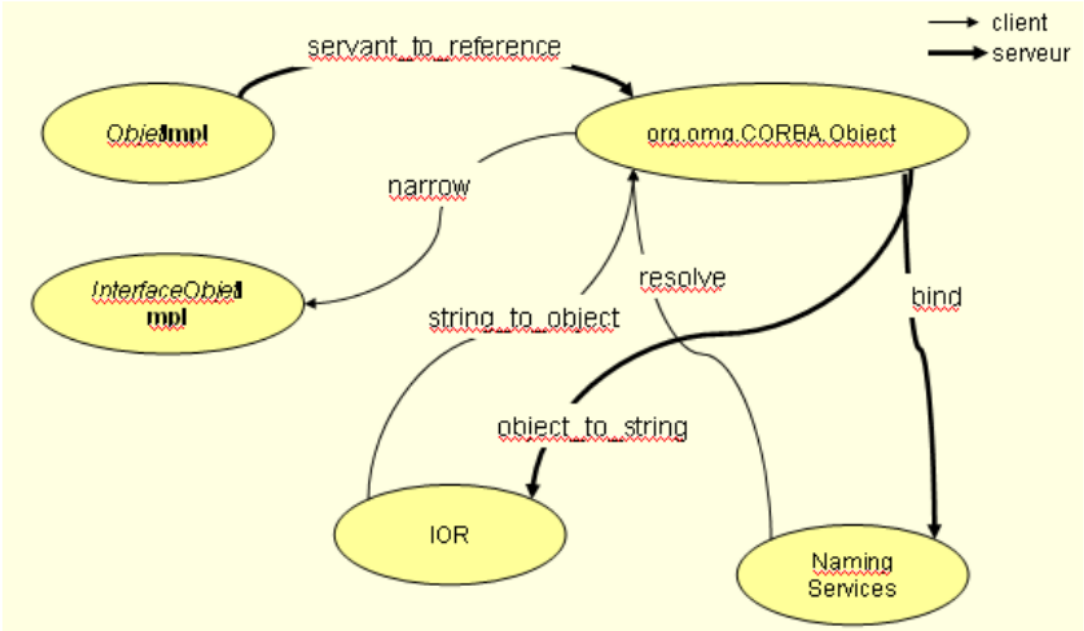
Soit deux machines A et B, on crée sur A un objet distribué OA, on crée sur B un objet distribué OB. Les deux objets distribués s'appellent mutuellement des méthodes distantes. Il est possible de :		Q 59
1	on crée un seul adaptateur sur A, et OA et OB s'enregistrent dans cet adaptateur	
2	on crée un adaptateur sur A, un autre sur B, et OA s'enregistre dans l'adaptateur de A, et OB s'enregistre dans l'adaptateur de B	X
3	on crée deux adaptateurs sur A, et OA s'enregistre dans l'un et OB dans l'autre	

En RMI, l'interface d'un objet distribué définit		Q 60
1	les prototypes de toutes les méthodes de l'objet distribué	
2	les prototypes des méthodes distantes de l'objet distribué	X

2. Questions libres (20 points)

Chaque question est notée sur 4 points.

Commentez le schéma suivant : **Q 1**



Ce schéma met en évidence les différents éléments (données, services, …) majeurs de l'architecture CORBA utilisé pour la mise en œuvre dans la création des objets distribués et l'utilisation de ces derniers.
Ces éléments sont représentés dans les bulles.
Les flèches correspondent aux transformations de ces éléments entre eux, et la nature des flèches au nom de la méthode qui réalise cette transformation.
Il y a deux circuits : celui réalisé par le client et celui réalisé par le serveur.

Pour le client, on lit sur le schéma que pour obtenir un « InterfaceObjetImpl », c'est à dire une connexion sur un objet distribué, il faut faire un « narrow » sur un objet CORBA. Cet objet CORBA peut être obtenu de 2 façons différentes :
- soit on n'a pas de service de nommage. Il faut alors l'obtenir à partir d'un IOR en utilisant la méthode « string_to_object »,
- soit on a un service de nommage et il faut l'obtenir à partir du nom de l'objet distribué enregistré, en utilisant la méthode « resolve ».

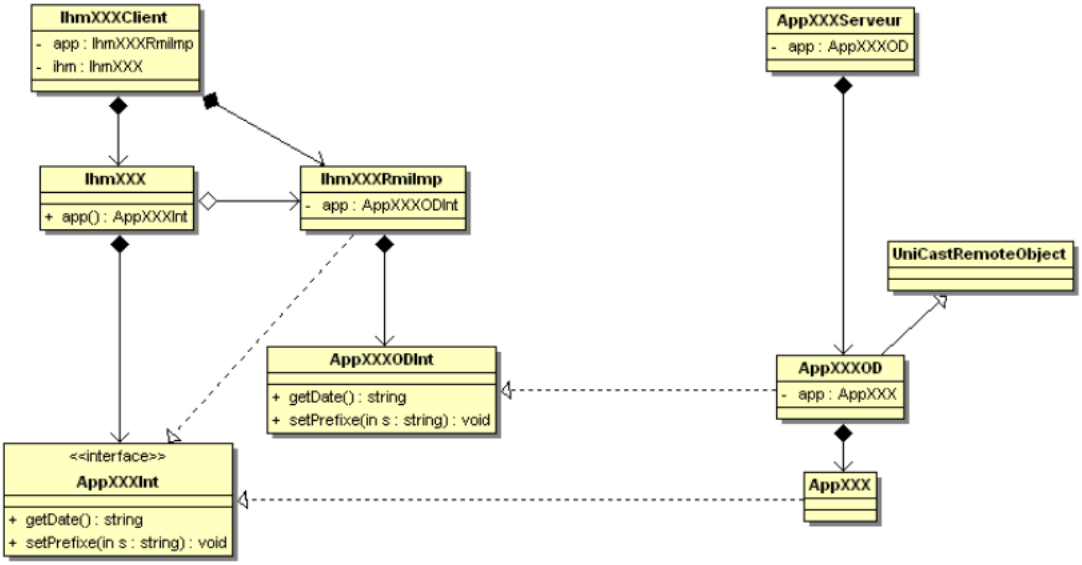
Pour le serveur, on lit sur le schéma que pour créer un objet distribué, il faut créer un servant à partir d'un ObjetImpl (qui hérite de ObjetImplPOA) en utilisant la méthode « servant_to_reference ». On obtient alors un objet CORBA. Il y a ensuite deux façons différentes :
- soit on n'a pas de service de nommage. Alors, on obtient l' IOR de l'objet CORBA en utilisant la méthode « object_to_string ».
- soit on a un service de nommage. Alors on enregistre l'objet CORBA dans le service de nommage avec la méthode « bind ».

A quoi sert le service de nommage dans une architecture CORBA ? **Q 2**

Le service de nommage est utilisé, dans une architecture CORBA, par le serveur, pour enregistrer, un « servant » (objet distribué) sous un nom logique.
Il est utilisé par les clients, pour localiser l'objet distribué sur le réseau et pour obtenir un canal de communication avec cet objet distribué et donc appeler les méthodes distantes.

En résumé, le service de nommage sert d'annuaire à tous les objets distribués d'un réseau donné.

Commentez le schéma suivant : **Q 3**



Ce schéma représente l'architecture UML d'un projet Java dans lequel l'IHM et l'applicatif sont séparé par une interface. C'est-à-dire que l'IHM est indépendante de l'implémentation de son applicatif.
L'IHM utilise les méthodes de l'applicatif à travers la description de son interface. Cet applicatif établit une communication RMI avec un OD (AppXXXOD).

La classe IhmXXXClient crée les deux objets : IhmXXXRmiImp et IhmXXX. Le 1er est passé en paramètre du deuxième.
IhmXXX utilise l'interface AppXXXInt .
IhmXXXRmiImp implémente l'interface AppXXXInt, et utilise l'interface AppXXXODInt (sur lookup).
Le serveur est la classe AppXXXServeur qui crée un OD AppXXXOD qui encapsule un applicatif metier : AppXXX. La classe AppXXX implémente l'interface AppXXXInt.
La classe APpXXXOD hérite de la classe UniCastRemoteObject.

Expliquez comment il est possible de créer de nouveaux objets distribués de manière distante. **Q 4**

Il est possible de créer de nouveaux objets distribués de manière distante, en créant un nouvel objet distribué qui sert de Factory, c'est-à-dire qui peu créer à son tort d'autres objets distribués.
Ce factory a donc au moins une méthode distante permettant de créer un autre objet distribué sur sa machine.

On est obligé d'utiliser un « factory » car un objet distribué est avant tout un objet créé en mémoire d'un programme. Il faut donc que ce programme existe avant toute chose, que ce programme soit sur une autre machine du réseau et que ce programme soit utilisable à distance. Quoi de mieux qu'un objet distribué pour créé un tel programme.

Citez les avantages des systèmes répartis. **Q 5**

Les avantages des systèmes répartis (1/2)

- Partage et Mise à disposition
  - partager des ressources et des services disponibles (ex: les systèmes d'exploitation répartis qui permettent de mettre en place un service de gestion de fichiers partagés et répartis)
- Répartition géographique
  - mettre à disposition des usagers les moyens informatiques locaux en même temps que ceux distants de leurs collègues (ex: un système de réservation d'hôtel répartis en différents pays, compagnies ou agences; autre ex: un système bancaire avec ses agences régionales et son siège social)
- Puissance de calcul :
  - paralléliser les algorithmes de calcul avec des environnements d'exécution spécifique comme PVM ou MPI)
- Disponibilité d'un service
  - continuer un service globalement même dégradé
  - exemple courant: la réplication d'un même service par l'installation de plusieurs serveurs équivalents (attention si les serveurs sont à états rémanents => réplication)

Les avantages des systèmes répartis (2/2)

- Flexibilité
  - par nature modulaire
  - continuité de service pendant la maintenance (remplacement d'un nœud)
  - l'informatique nomade : portable et points d'accès mobiles sur un réseau réparti aux frontières floues (Internet)
  - les problèmes posés sont :
    - la localisation
    - l'identification
    - l'authentification
    - l'optimisation des temps de connexion
- Adaptabilité à une forte croissance des besoins informatiques d'une entreprise
- => une meilleure définition :
  - **La répartition est la mise à disposition d'un ensemble de ressources et services connectés via un réseau pour tous les usagers qui possèdent un droit d'accès en un point quelconque.**

*Vous répondez à ces questions sur une **copie vierge** en mettant bien le numéro de la question, sans oublier votre nom et prénom.*