Maîtrise Informatique
M5: Programmation Répartie et Architecture N 1/3

DS – Session de Juin 2003

Denis Caromel — Fabrice Huet

Professeur
Université de Nice - Sophia Antipolis, Département Informatique

Les documents et calculatrices sont autorisés.
Prenez le temps de bien lire et de bien comprendre les questions !
Les réponses ne sont pas longues, soyez *concis* et ... *lisible* !
Notez *clairement* sur votre feuille les numéros des questions (exemple: 3. b.).
Durée: 2H00

1 (3 points)

Une application client-serveur est composée des classes/interfaces suivantes:
Serveur: interface du serveur distant
ServeurImpl: implémentation du serveur
Client: interface du client
ClientImpl: implémentation du client
Dans un premier temps, nous n'allons considérer que des communications distantes allant du client vers le serveur:

a.) Indiquez la ou les interfaces distantes ("'Remote Interface"). b.) indiquez la ou les classes RMI pour lesquelles vous allez générer un stub et un skeleton . c.) Si maintenant nous voulons avoir la possibilité de faire un call-back, quelles sont les modifications à apporter ?
Bien lister ces modifications au niveau des Interfaces, des Classes et du code source.

2 (3 points)

La compagnie VenToo veut mettre en place un système de vente de produits en ligne. Sa cible principale est le grand public mais elle veut aussi que d'autres vendeurs puissent accéder aux objets qu'elle vend (pour par exemple faire de la revente). Cela implique d'inclure les objets en question dans des catalogues, qui sont en fait des logiciels qui dans certains cas existent déjà, certains ayant même été réalisés dans les années 1980.

a.) Dessinez une architecture permettant de mettre en place ce double service et justifiez-la.

b.) Quelles plateformes et langages conseilleriez vous à une entreprise qui veut construire un tel service sur le web ?
Donner une reponse éventuellement en analysant l'existant.

3 (4 points)

a.) Pourquoi a-t-on toujours besoin de stub avec RMI Java 2, alors que les skeleton ne sont plus necessaires? Bien expliquer la différence, au niveau conceptuel et pratique.

b.) Donner le principe du code d'un skeleton spécifique de Java RMI Version 1, et le code générique remplacant ce code dans Java RMI Version 2.

4 (6 points)

a.) Faites un dessin et expliquez de façon synthétique le principe du Ramasse Miettes de RMI (RMR).

b.) Quelles sont les relations entre le ramasse miettes RMI et le ramasse miette standard Java?

c.) Que se passe-t-il si on augmente la durée du "lease"? Avantages? Inconvévients?

d.) Etant donné que le RMR fonctionne avec des "leases", étant donné qu'après un certain temps une référence sera annulée, quel est l'avantage d'avoir tout de même du "reference counting"? Pensez à l'aspect réactivité pour répondre à cette question.

e.) Est-ce que cet algorithme de ramasse miettes réparti permet de gérer les cycles? Pourquoi?

5 (2 points)

Quels sont les deux types d'intéropérabilité offerts par CORBA?

6 (2 points)

Lors du déploiement de la partie cliente d'une application Java-RMI, vous vous apercevez que vous ne pouvez pas faire fonctionner le téléchargement dynamique de classe: il est impossible pour votre client de télécharger la classe ServeurImpl_Stub (et seulement celle-là pour simplifier le problème).
Que pouvez vous faire, sachant que vous ne pouvez pas vous connecter sur le serveur pour récupérer la classe du stub?