

Intergiciels

Examen

Daniel Hagimont

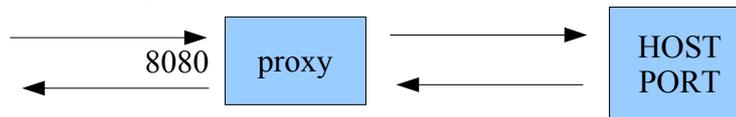
Durée: 2 heures, documents autorisés

Lire l'ensemble des énoncés avant de commencer à répondre. La **clarté**, la **précision** et la **concision** des réponses, ainsi que leur **présentation matérielle**, seront des éléments importants d'appréciation. Donnez essentiellement les programmes Java demandés. Dans vos programmes, vous n'avez pas à programmer les imports et le traitement des exceptions.

PROBLÈME I (Sockets)

On veut implanter avec des sockets en Java un proxy de streaming TCP. Plus précisément, vous devez réaliser un programme qui :

- accepte des connexions TCP sur le port 8080 (soit C1 une telle connexion)
- pour chaque connexion acceptée, crée une connexion (soit C2 cette connexion) vers un site donné (par 2 constantes HOST, PORT)
- toute donnée reçue sur C1 est envoyée sur C2
- toute donnée reçue sur C2 est envoyée sur C1
- on ne peut faire aucune hypothèse sur le volume ou la vitesse de transmission des données sur ces connexions TCP. Les données peuvent être envoyées dans un sens, dans l'autre, ou dans les deux sens.



PROBLÈME II (RMI)

Le rmiregistry fournit par java permet de faire un lookup à distance, mais pas de faire un bind à distance (le programme faisant le bind doit être sur la même machine que le rmiregistry).

Pour corriger cela, on vous demande de faire un nouveau registry implanté à l'aide de RMI (et de son rmiregistry).

Votre registry doit fournir une classe MyNaming qui fournit les méthodes :

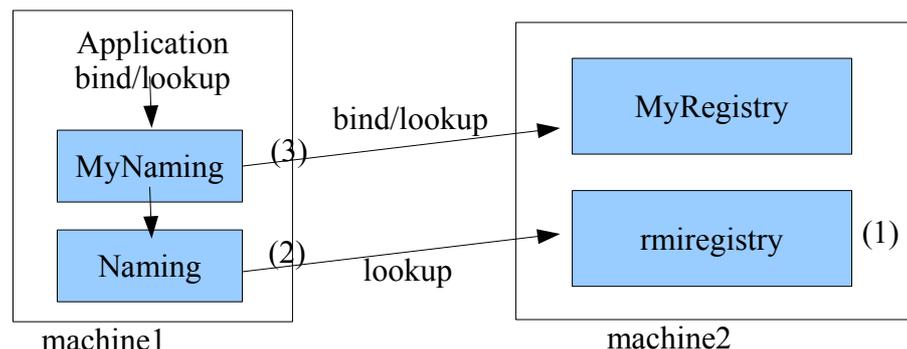
- public static void bind(String host, int port, String name, Serializable stub);
- public static Serializable lookup(String host, int port, String name) ;

On peut noter qu'un stub de RMI est un objet sérialisable. Cette class MyNaming est utilisée par l'utilisateur du nouveau registry.

La classe MyNaming utilise un serveur RMI qui a été lancé sur une machine distante. Ce serveur est d'interface MyRegistry et de classe MyRegistryImpl. Le serveur est lancé sur la machine distante et il s'enregistre auprès d'un rmiregistry lancé sur cette même machine (1). Le serveur stocke (lors d'un bind) les stubs dans une HashTable<String, Serializable> et les retourne (lors d'un lookup).

Evidemment, la classe MyNaming utilise le rmiregistry (et la classe Naming) pour récupérer la référence de l'instance de la classe MyRegistryImpl qui a été créée. Il peut ensuite utiliser les méthodes statiques ci-dessus pour enregistrer à distance (bind) une référence dans MyRegistry (3).

Vous devez donc fournir le code de l'interface MyRegistry et des classes RegistryImpl et MyNaming.



PROBLEME III (JEE)

Les réponses doivent être rédigées dans les cadres sur cette feuille. Être concis pour tenir dans le cadre.

Question 1

Expliquer en quelques mots l'utilité du paramètre d'annotation `fetch=FetchType.EAGER`

.....

Question 2

Si en JPA, on dispose de deux Entity Person et Address

<pre>@Entity public class Person { @Id String name; @OneToMany Collection<Address> addresses; ... }</pre>	<pre>@Entity public class Address { @Id String place; String town; }</pre>
---	--

Dans une facade, si j'écris :

```
Person p1 = em.find(Person.class, "tchana");
Person p2 = em.find(Person.class, "hagi");
Address a = em.find(Address.class, "enseeiht");
p1.getAddresses().add(a);
p2.getAddresses().add(a);
```

En fait, je veux programmer que tchana et hagi sont tous les 2 à l'enseeiht. Ce code est il correct ?
Que faut il ajouter/modifier pour qu'il le soit ?

.....

Question 3

Avec les Entity précédents, j'ai une servlet qui a récupéré la saisie d'une nouvelle adresse pour une personne et qui appelle une facade qui propose la méthode suivante :

```
public void addAddress(String person_name, Address a) {
    Person p = em.find(Person.class, person_name);
    p.getAddresses().add(a);
}
```

Quels problèmes peut poser la méthode ainsi codée ? Expliquez et proposez une correction.

.....
