

Examen – 2^{nde} session

Conception d'applications réparties (CAR)

Tous documents papier autorisés. Téléphones, calculatrices et ordinateurs interdits.
Le barème est donné à titre indicatif. 3 heures.
Ce sujet comporte 2 pages.

1 Questions de cours (2 points)

- 1.1 Qu'est-ce qu'une référence d'objet distante ?
- 1.2 Dans un mécanisme d'invocation de méthode distante, quel est le rôle d'une souche cliente ?

2 REST (3 points)

On s'intéresse à une architecture répartie de type RESTful dont les ressources sont les commandes gérées par un site de commerce en ligne.

- 2.1 Que vont permettre de faire les quatre opérations *Create Read Update Delete* associées au principe REST dans cette architecture ? (1 point)
- 2.2 En présence de concurrence, par exemple avec un serveur multi-threadé, quelle politique de synchronisation proposez-vous de mettre en œuvre pour les quatre opérations *Create Read Update Delete* ? Indiquer le rôle joué par chaque opération dans la politique de synchronisation. (1 point)
- 2.3 Au delà des aspects liés à l'implantation (HTTP, XML, HTML, JSON, etc.), quelle est la différence principale entre une architecture RESTful et une architecture répartie utilisant SOAP ? (1 point)

3 Serveur de fichiers (5 points)

On souhaite mettre en place un serveur Java RMI permettant de transférer des fichiers selon les mêmes principes que FTP : connexion avec login/mot de passe, *upload/download* de fichier, changement de répertoire, fermeture de connexion. Afin de limiter les invocations avec des volumes trop importants de données, on souhaite par ailleurs que les *upload/download* se fassent par blocs de 1024 octets. Plusieurs transferts, éventuellement entre le même client et le même serveur, doivent pouvoir se dérouler simultanément.

- 3.1 Proposer une interface RMI permettant de faire cela. (1 point)

- 3.2 Écrire en Java ou en pseudo-code le comportement du serveur. (3 points)
- 3.3 On souhaite que le client puisse détecter les pannes du serveur. Expliquer en français les modifications que vous proposez d'effectuer pour cela dans le code précédent. (1 point)

4 Annuaire de services (6 points)

Soit un Annuaire de services qui enregistre des associations entre une référence et un nom de service. Soit un Hébergeur de services qui fournit des services et les enregistre auprès de l'Annuaire. Soit un Consommateur de services qui interroge l'annuaire pour retrouver la référence d'un service à partir d'un nom.

- 4.1 Expliquer en français quelles sont les deux méthodes principales que vous proposez pour l'interface de l'Annuaire. Justifier et expliquer le choix des paramètres de ces méthodes. (1 point)
- 4.2 Décrire en français le scénario de fonctionnement entre l'Annuaire, un Hébergeur et un Consommateur de services. (1 point)
- 4.3 En vous inspirant de WSDL, proposer quatre modes d'interaction entre un Consommateur et un Hébergeur de services. (1 point)
- 4.4 Définir en français un protocole (messages et format de ces messages) utilisant TCP et permettant de mettre en œuvre ces quatre modes d'interaction. (1 point)
- 4.5 Écrire en Java ou en pseudo-code le comportement d'un Hébergeur pouvant fonctionner avec ces quatre modes d'interaction. (2 points)

5 Méthode de rappel en RMI (4 points)

On considère deux objets RMI X et Y. Y fournit une méthode `inversion` permettant de faire l'inversion d'une matrice. Une matrice est un tableau bidimensionnel de réels double. On considère que la méthode `inversion` effectue une itération sur le nombre de lignes de la matrice (on ne demande pas de fournir plus de détails sur le code de cette méthode). On souhaite qu'à chaque pas de l'itération, l'objet Y fournisse à l'objet X l'indice de l'itération en invoquant une méthode `callback` que vous définirez. X affichera l'indice reçu. Par ailleurs, X fournit une méthode `main` qui invoque la méthode `inversion` en transmettant une matrice que l'on suppose connue.

- 5.1 Proposer le code Java de l'interface `XItf` de l'objet RMI X ainsi que de la classe `XImpl` implantant cette interface. Même question pour l'interface `YItf` de l'objet RMI Y et de sa classe d'implantation `YImpl`. (2 points)
- 5.2 Que se passerait-il si l'objet X n'était pas *multi-threadé* ? (1 point)
- 5.3 On souhaite que l'invocation de la méthode `callback` soit asynchrone. Java RMI propose-t-il cette notion ? Si oui, comment ? Si non, proposez une solution de remplacement ? (1 point)