Agenda Répartie

Giuseppe Lipari

Cet exercice porte sur la mise en œuvre d'un système d'agendas répartis, multi-utilisateurs et multi-thread. Soient des objets `Agenda` et des objets `Utilisateur` représentant respectivement des agendas et des utilisateurs de ces agendas. Les objets `Agenda` stockent des rendez-vous, représentés par des objets `RDV`, et comprennent un identifiant unique (entier), une date (`java.util.Date`), une heure de début (entier) et une heure de fin (entier). Un agenda fournit quatre services permettant respectivement de créer, modifier, lire et détruire un rendez-vous. Les `Agenda` sont des objets accessibles à distance en Java RMI. Les `RDV` sont des objets transmis par copie.

1 Question

Proposer une interface Java RMI `AgendaItf` pour les objets `Agenda` et proposer une classe pour les objets `RDV`. Expliquer en français vos choix de conception en rapport avec l'écriture de cette interface et de cette classe.

2 Question

Dans le cas où ces quatre services sont accessibles via REST, quelles ressources proposez vous de considérer ? Quelles actions proposez vous pour ces ressources ?

3 Question

On souhaite pouvoir lister les rendez-vous stockés dans un `Agenda`. Pour cela, un service `list` est défini qui retourne un itérateur. L'itérateur permet de récupérer à distance et un à un les rendez-vous stockés dans un `Agenda`. Chaque itérateur fournit les services `hasNext` (aucun paramètre, retourne un booléen qui vaut vrai si il y a un rendez-vous suivant, faux sinon) et `next` (aucun paramètre, retourne le rendez-vous suivant, ou null si il n'y en a pas).

Proposer une ou plusieurs interfaces Java RMI permettant de mettre en oeuvre cette fonctionnalité.

4 Question

On souhaite pouvoir inviter des utilisateurs à un rendez-vous. Chaque utilisateur doit pouvoir à tout moment notifier son acceptation ou son refus de participer au rendez-vous. Il doit pouvoir également changer d'avis. Enfin, si un rendez-vous est modifié ou détruit, les utilisateurs l'ayant accepté doivent être prévenus.

Proposer une ou plusieurs interfaces Java RMI permettant de mettre en œuvre ces fonctionnalités.

5 Question

Dessiner un diagramme de séquence qui montre les échanges de messages entre les objets dans le scénario suivant :

— L'utilisateur Amélie crée un rendez-vous ;
— Il invite l'utilisateur Bruno à cet rendez-vous
— Bruno accepte l'invitation
— Amélie change d'avis et change le rendez-vous : par conséquent, Bruno est notifié du changement ;

Remarquez les appelles de méthodes locales et les appelles distants.

6 Question

Écrire la classe Java RMI implémentant un `Agenda`. Le corps des méthodes peut être écrit en pseudo-code.

7 Question

Maintenant, on suppose de vouloir faire la même chose avec une application Web en utilisant la technologie REST.

Comment on peut mettre en place un service de notification pour l'utilisateur ?