

Bureau de vote

Problème

On désire réaliser un service de vote électronique sur internet. Ce service permet à des utilisateurs de donner leur avis sur des questions soumises par d'autres utilisateurs. Pour voter, les votants s'inscrivent dans des bureaux de votes virtuels. Les questions sont aussi envoyées au bureau de votes qui se charge alors de soumettre celles-ci au vote de tous les votants inscrits. Donc, un bureau de votes gère une liste des votants inscrits et une liste des questions en cours de vote. Lorsqu'un votant répond à une question, il a seulement les choix suivants : OUI, NON ou ABSTENTION. Lorsque le vote d'une question est terminé, le bureau de votes publie le résultat du vote sur un tableau d'affichage des résultats de votes. Un tableau d'affichage peut contenir les résultats de différents bureaux de votes. De plus, il peut y avoir différents tableaux d'affichage.

Ce service est réalisé par la coopération / communication de différents processus : des serveurs gérant les bureaux de votes et les tableaux d'affichage et des processus pour les utilisateurs permettant de voter, de soumettre une question, de consulter les résultats, etc.

On dispose de la classe suivante :

```
package serviceVote;
class ResultatVote implements serializable {
    public String identifiant_question;
    public String question;
    public long nb_votants;
    public long nb_abstentions;
    public long nb_oui;
    public long nb_non;
    public ResultatVote(String id, String q, ...) {...};
}
```

Cette classe représente le résultat d'un vote : un identifiant unique désignant la question, une chaîne pour le texte de la question, le nombre de votants ayant répondu, le nombre d'abstentions, de votes oui et de votes non.

Le tableau d'affichage

Le tableau d'affichage est défini par une interface permettant d'enregistrer un résultat de vote, d'obtenir le résultat associé à un identifiant de question et d'obtenir la liste des identifiants de question.

Le bureau de vote

Un bureau de votes possède une opération permettant de retrouver le tableau d'affichage associé à ce bureau, une pour soumettre une question au vote des votants inscrits, une opération d'inscription pour un votant et l'opération pour retirer un votant de la liste des votants.

Le votant

L'interface `Votant` représente un utilisateur désirant voter. L'opération `recevoirQuestion` permet d'informer l'utilisateur d'une nouvelle question (id, question) et lui indique le bulletin pour déposer son vote.

Le bulletin

L'interface `Bulletin` a une opération `voter` pour recevoir un vote anonyme d'un votant. Attention, il ne peut voter qu'une seule fois. L'opération transmettra ce vote à l'Urne qui réalise le vote.

L'urne

Elle récolte les différents votes pour une question donnée et transmet le résultat bureau quand le vote est terminé.

Question 1

Donnez les différentes interfaces RMI permettant de décrire cette application.

Question 2

Donnez le code de la classe `UrneImpl` qui implémente le service RMI `Urne`.

Question 3

Donnez le code de la classe `BureauVoteImpl` qui implémente le service RMI `BureauVote`.

Question 4

Donnez le code du processus serveur abritant un bureau de vote qui sera enregistré auprès du service de nommage RMI. En paramètre de la ligne de commande seront donnés le nom sous lequel sera enregistré le bureau de vote et le nom sous lequel est enregistré le tableau d'affichage que le bureau utilisera.

Question 5

Implanter en Java un client permettant d'afficher le résultat du vote d'identifiant `MaQuestion` du bureau de vote enregistré dans le service de nommage RMI sous le nom `MonBureau`.

Question 6

Qu'apporte l'utilisation de RMI pour la conception et la réalisation d'une telle application par rapport à l'utilisation directe des sockets ?

Question 7

On désire rendre accessible l'application par un client léger, le choix se portant sur un navigateur WEB. Proposez l'architecture de l'application (les différents processus, les protocoles utilisés pour communiquer, ...) ainsi augmentée de son accès par le WEB.