GIS-IMA 3ème année, Devoir Surveillé Architectures Logicielles

Durée 2 heures, tous documents papiers autorisés

1 Programmation par composants Java Beans (8 points)

On désire modéliser très simplement la gestion d'actions en bourse par différents porteurs. Vous disposez du composant Java Beans `PorteurBean` qui dispose, en particulier, de méthodes de ventes et d'achats d'actions :

```
public class PorteurBean {
    ...
    public void vendre(ActionBean a) { ... }
    public void acheter(ActionBean a) { ... }
}
```

Une action sera modélisée par le composant Java Beans `ActionBean` qui dispose des caractéristiques suivantes :
– une propriété *nom* de type `String` pour désigner le nom de l'action,
– une propriété **liée** *cours* de type `double` pour connaître le cours de l'action.
Question 1.1 : écrire le composant `ActionBean` qui est totalement indépendant du composant `PorteurBean`.

On désire maintenant faire en sorte que les porteurs soient à l'écoute de certaines actions et effectuer automatiquement des opérations d'achat et/ou vente sur ces actions lorsqu'elles dépassent un seuil fixé : si le cours est inférieur au seuilMin alors achat, si le cours > seuilMax alors vente.
Par exemple, un porteur p1 s'intéresse à l'action a1 de nom 'Planet Pizza', désire acheter si le cours < 25 et vendre si le cours > 45.
Question 1.2 : Sans modifier le code des composants `PorteurBean` et `ActionBean`, écrire la classe `PorteurAdaptateur` qui permet de relier ces deux composants, de gérer des seuils données et d'assurer l'automatisation des opérations d'achats et vente en fonction de ces seuils.
Question 1.3 : écrire l'extrait de code Java qui, pour des variables p1 de type `PorteurBean` et a1 de type `ActionBean` configure et relie ces objets **déjà instanciés** grâce à un objet de type `PorteurAdaptateur` pour répondre à l'exemple ci-dessus.

2 Architecture logicielle distribuée (12 points)

On désire concevoir une architecture distribuée qui permet de réaliser une élection présidentielle entièrement par bulletins électroniques. Cette élection se déroule en trois périodes de temps successives :

Phase d'inscription les électeurs et/ou futurs électeurs ont la possibilité de s'inscrire à un bureau de vote de leur choix avant une date donnée, chaque bureau de vote a un nom qui permet de l'identifier. Chaque électeur validé par l'administration recevra un équivalent de login/password pour la phase d'élection. Plus précisément, le futur électeur doit fournir les renseignements de nom, prénom, date de naissance, et nom de la commune de naissance. A l'issue de la fourniture de ces informations au bureau de vote, ce dernier fournit tout de suite un code. Ce code sera exigé pour récupérer quelques jours plus tard les login et mot de passe nécessaires pour la phase suivante. Ce login/password ne sera délivré que si le bureau de vote a validé l'inscription.

Phase d'élection Durant une période donnée, les électeurs validés peuvent voter une seule fois à distance.

Phase de dépouillement Le ministère de l'intérieur a la charge de collecter les résultats et de proclamer le vainqueur de ce suffrage universel. Il n'y a pas de second tour, celui qui a le plus de voix l'emporte.

Pour la suite, on utilisera la technologie J2EE-EJB pour mettre en place cette architecture logicielle, on ne se préoccupera ni des composants web, ni de la manière dont on gère la persistance des informations mais uniquement des fabriques et interfaces des composants EJB accessibles à distance
Toutes les démarches (inscription, élection, dépouillement) se font à distance par Internet. La gestion des électeurs n'est pas centralisée. Pour chaque bureau de vote, il existe une application J2EE qui gère ses électeurs.
Les électeurs disposent d'un unique portail Web pour exécuter les phases d'inscription et d'élection. A charge de ces composants webs sur ce portail, d'accéder aux composants EJB des différents bureaux de vote.

Question 2.1 : Proposez une architecture logicielle qui répond à ce problème. Indiquez le ou les services de noms nécessaires, le ou les différents composants EJB nécessaires. Décrire le déploiement des différents composants logiciels et applications sur les différents serveurs. Ne décrire pour chaque composant introduit que les éléments accessibles (fabrique et interface). Utilisez la norme EJB pour spécifier ces interfaces Java (ne pas mettre les lignes import), proposez des exceptions aux différentes méthodes pour garantir un comportement le plus cohérent possible.

Question 2.2 : Ecrire le code Java exécuté au niveau du ministère de l'intérieur qui permet de connaître le nombre de votes pour chaque candidat. On pourra utiliser (ce n'est pas une obligation), la méthode Collection list(String JNDIDir) du serveur de noms qui permet de récupérer une collection de noms enregistrés sur ce serveur dans le répertoire passé en paramètre. Attention, ce sont juste les noms symboliques, pas les adresses des références distantes. Ainsi, l'extrait de code suivant permet d'afficher tous les noms symboliques JNDI enregistrés sur un serveur de noms dans le répertoire '/ocaron/'.

```
jndi=InitialContext() ;
Collection lesNomsEnregistres=jndi.list(''ocaron/'') ;
for (Iterator it=lesNomsEnregistres.iterator();it.hasNext();) {
  System.out.println(''nom JNDI : ''+ ((String) it.next())) ;
}
```