Programmation Objet

DESS Génie Logiciel, janvier 2003
Tout document autorisé, durée 3 heures

Justifiez clairement et succinctement vos choix. Lorsque plusieurs alternatives sont possibles, présentez-les en considérant les avantages et les inconvénients.

Quand vous utilisez un pattern, nommez-le.

Toute réponse difficile à comprendre sera considérée comme fausse.

Si vous avez un doute concernant la syntaxe de Java ou d'UML, ou sur l'utilisation d'une classe de l'API Java, précisez le clairement sur votre copie.

Vous pouvez utiliser les diagrammes UML (classes et séquences) à chaque fois que vous le jugerez nécessaire.

Excercice 1

On considère trois classes `C1`, `C2` et `C3` qui implémentent respectivement les méthodes `m1`, `m2` et `m3`. La méthode `m2` est utilisée dans `m1` à travers un objet `x`. La méthode `m3` offre un service similaire à `m2`. Pour simplifier, `m2` et `m3` n'ont pas de valeur de retour.

```
... class C1 {
     ...
     ... m1(...) {... x.m2(...); ...}
     ...
}

... class C2 {
     ...
     ... void m2(...) {...}
     ...
}

... class C3 {
     ...
     ... void m3(...) {...}
     ...
}
```

Expliquer comment on peut remplacer l'usage de `m2` par celui de `m3` en fonction des différents cas possibles. On traitera en particulier les critères suivants (certaines combinaisons sont bien sûr impossibles) :

- possibilité ou non de modifier les classes `C1`, `C2` et/ou `C3`, ainsi que leurs clients,
- accessibilité (`public` ou autre) des classes `C1`, `C2` et `C3` et des méthodes `m1`, `m2` et `m3`.
- choix du remplacement de `m2` par `m3` à la compilation ou à l'exécution, définitif ou reversible,
- signature de `m2` et `m3` égales ou différentes,
- déclaration de `x` comme variable d'instance de `C1` ou dans les paramètres de `m1`,
- type déclaré de `x` égal à `C2` ou à une classe héritée par `C2` ou à une interface implémentée par `C2`.

Prenez soin dans chaque cas de présenter les différentes alternatives possibles, en les illustrant par des schémas UML et/ou du code Java et en précisant dans quels cas il convient d'utiliser plutôt l'une que l'autre.

Excercice 2

Le but de cet excercice est surtout de proposer une architecture. Il est inutile de trop développer le code lorsque celui-ci vous paraît évident.
Lorsque vous utilisez un pattern, nommez-le.
Le choix de l'architecture dépend parfois de critères qui ne sont pas précisés dans la question. Donnez alors les alternatives et dans quels cas il convient de choisir l'une ou l'autre.

On s'intéresse dans un premier temps à un automate dont les transitions sont étiquetés par des entiers. Les transitions peuvent donc être stockées sous la forme d'une matrice dont les lignes sont indicées par les états et les colonnes par les étiquettes.

Question 1 On stocke la matrice des transitions elle-même. Selon que la matrice soit creuse ou non, on utilisera deux structures de stockage différentes. Proposer une architecture en conséquence.

Question 2 La matrice des transitions est maintenant éclatée sur chaque état. On peut donc stocker les transitions à partir de chaque état différemment, par exemple sous la forme d'un vecteur ou d'une liste de couples (étiquette,destination) selon que le nombre de transitions partant de cet état soit élevé ou non. Proposer une architecture en conséquence.

Question 3 Lors de la lecture d'une suite de nombre, on parcourt l'automate. Le résultat du parcours sera le numéro de l'état atteint, ou une exception si à un moment donné, la lecture d'un entier ne correspond à aucune transition de l'état courant. Implémenter cette fonctionnalité.

Question 4 Les transitions sont maintenant étiquetés par des chaînes de caractères. Implémenter cette nouvelle fonctionnalité en réutilisant l'architecture de la question 3.

Question 5 On suppose que les étiquettes associées aux transitions puissent être transmises par le client avant chaque parcours. Implémenter cette nouvelle fonctionnalité.

Question 6 On souhaite associer à chaque transition une action à effectuer lors de son parcours. Rajouter cette fonctionnalité.

Question 7 On suppose maintenant qu'un même automate puisse être partagé par plusieurs clients, chacun souhaitant lui associer son propre ensemble d'actions indépendemment des autres. Implémenter cette nouvelle fonctionnalité.

Question 8 On suppose que la lecture d'une séquence de mots se concrétise par une suite de notes jouées par un ou plusieurs instruments simultanément. Utiliser l'architecture de la question 7 pour réaliser cette application