

Conception et Programmation Objet Avancées

- DOCUMENTS AUTORISÉS
- [ex0], [ex1, ex2], [ex3, ex4] SONT A RENDRE SUR DES COPIES SEPARÉES

Exercice 0 (Questions de cours) (2 pts)

Question 1 • Quelle est la différence entre le pattern *Décorateur* et le pattern *Stratégie* ? donnez un exemple. (1 pts)

Question 2 • Expliquez la différence qui existe entre la version "uniforme" et la version "TypeSafe" du pattern *Composite*. (1 pts)

Exercice 1 (6 pts)

Les clients d'une banque sont classés en deux catégories : Ceux qui ont le droit au crédit ; Ceux qui n'ont pas ce droit. Lors de la demande d'une carte de paiement, les premiers reçoivent une carte de crédit (à débit différé sur leur compte) alors que les seconds peuvent seulement avoir une carte de débit (à débit immédiat sur leur compte).

Question 1 • Quel pattern de conception permet-il de modéliser la création de la carte de paiement en fonction du client ? (0,5 pts)

Question 2 • Modélisez son utilisation par un diagramme de classes. (2 pts)

Il existe deux modèles de cartes de débit et de crédit, à savoir les cartes VISA et les cartes MASTERCARD.

Question 3 • Quel est le pattern à utiliser pour cette nouvelle situation ? (0,5 pts)

Question 4 • Modélisez, à l'aide d'un diagramme de classes, la création d'une carte de paiement en fonction de sa famille (crédit/débit). (3 pts)

Exercice 2 (4 pts)

Nous disposons d'une classe **Contact**, qui possède entre autres une méthode **envoyer(Message)**. Nous souhaitons enrichir l'application avec des listes de diffusion. Ces listes peuvent contenir des contacts et d'autres listes.

Question 1 • Quel pattern de conception permet-il de modéliser une hiérarchie de classes permettant d'envoyer un message aussi bien à un contact individuel qu'à une liste de diffusion (et donc à tous les contacts inclus dans celle-ci) ? (0,5 pts)

Question 2 • Modélisez son utilisation par un diagramme de classes. (3,5 pts)

Exercice 3 (4 pts)

On dispose d'une classe **Dictionnaire**. Elle propose la méthode **estValide(String mot)** : **boolean** qui vérifie l'existence d'un mot. Cette vérification se fait par lecture d'un fichier sur le disque, et consomme un temps considérable.

Question 1 • Quel pattern de conception permet de ne pas rechercher les mots qui ont été déjà vérifiés ? (0,5 pts)

Question 2 • Modélisez son utilisation par un diagramme de classes. (2,5 pts)

Question 2 • Ajoutez une note UML pour expliquer la réalisation de la méthode qui permet de ne pas rechercher un mot déjà vérifié. (1 pts)

Exercice 4 (4 pts)

La méthode **play()** d'un lecteur vidéo **VideoPlayer** peut lire des fichiers MP4. Les classes **VLCPlayer** et **MOVPlayer** peuvent respectivement lire des fichiers VLC et MOV avec les méthodes **vlcPlay()** et **movPlay()**.

Question 1 • On suppose que ces éléments ne sont pas modifiables (**VideoPlayer**, **VLCPlayer** et **MOVPlayer**). On souhaite qu'une classe cliente du **VideoPlayer** puisse, en appelant sa méthode **play()**, lire non seulement des fichiers MP4, mais également des fichiers VLC et MOV. Pour implémenter cette possibilité, quel pattern peut-on mettre en œuvre ? (0,5 pts)

Question 2 • Modélisez son utilisation par un diagramme de classes. (2,5 pts)

Question 3 • Ajoutez une note UML pour expliquer la réalisation de la méthode **play()**. (1 pts)