

**Examen en “Ingénierie Logicielle” - FMIN107E**  
**M1 Informatique (Master IFPRU)**  
**Partie A - Schémas de Conception et de Réutilisation.**

Christophe Dony, Clémentine Nebut

Janvier 2010

Durée prévue : 1h20 sur les deux heures. Documents de cours autorisés.

La précision et la concision des réponses sont notées, ainsi que la lisibilité.

## 1 Frameworks, Schémas de conception, Réutilisation

### 1.1 Locations, version 1

Considérons la modélisation objet d'un logiciel utilisé dans un magasin de location. Le magasin permet à ses abonnés de louer des produits : des cassettes VHS (dites “K7”) (prix base : 3 euros HT<sup>1</sup>, TVA<sup>2</sup> : 19%), des DVDs (prix base 2 euros HT, TVA : 19%) et des livres (le prix hors taxe de location d'un livre est ajusté chaque semaine en fonction de l'évolution du prix de vente du même livre dans le commerce, TVA : 5%).

Pour louer tout client doit posséder un compte sur lequel il dépose à l'avance une somme qui servira à payer les locations. Un compte est valable un an. Le prix de location dépend du produit et du client car chaque client gagne 10% de réduction sur chaque location par année d'ancienneté.

Voici un exemple d'utilisation d'instances des classes représentant les concepts précédents.

```
Client cl = new Client("Dupont", "Montpellier");
Compte cmt = new Compte(cl, "14/01/2008", 1000);
DVD lgv = new DVD("La grande vadrouille");
```

#### Questions.

1. Les instances<sup>3</sup> des classes `Livre`, `DVD`, `K7` doivent pouvoir recevoir les messages `prixHT()` et `prixTTC()` rendant respectivement leur prix hors taxe et leur prix TTC<sup>4</sup>.  
Définissez ces trois classes au sein d'une hiérarchie que vous construirez.
2. Dotez-les des attributs<sup>5</sup> et constructeurs utiles à l'énoncé.
3. Donnez la ou les définitions des méthodes `prixHT` et `prixTTC` sur les classes de votre hiérarchie où il est nécessaire qu'elles soient définies.
4. Définissez la classe `Compte` dotée des attributs induits par l'énoncé<sup>6</sup>.
5. Ajoutez ce qu'il faut à la classe `Compte` pour que chaque compte possède un numéro unique, alloué automatiquement au moment de sa création.
6. Le coût d'une location dépend du produit loué mais aussi de l'utilisateur qui loue puisque celui-ci peut avoir une réduction liée à son ancienneté.

Ecrivez la méthode `float prixFinalLocation(float prixCourant)` de la classe `Compte` qui reçoit en paramètre le prix TTC du produit et qui calcule le prix final de location en appliquant éventuellement la réduction que possède le titulaire du compte courant.

---

1. “Ht” means : with no tax included

2. VAT

3. Vous ne vous occuperez pas dans cet examen de distinguer produit et exemplaire de produit. Un DVD est simplement représenté par une instance de la classe `DVD`.

4. prix Toutes Taxes Comprises : Price with VAT included

5. N'écrivez aucun accesseur inutile.

6. L'ancienneté d'un client sera matérialisée uniquement par un attribut `réduction` et il n'est pas demandé d'écrire les méthodes qui modifient cet attribut. On supposera qu'il est correctement mis à jour.

7. Pour différentes raisons d'utilité pratique que vous pourrez imaginer, et aussi pour la beauté des choses, ou souhaite définir à la fois une méthode `float prixPourLocation (Compte)` sur la classe `Produit`<sup>7</sup> et une méthode `float prixPourLocation (Produit)` sur la classe `Compte`. Elle pourront être utilisées ainsi :

```
lgv.prixPourLocation(cmt);  
cmt.prixPourLocation(lgv);
```

Les deux expressions précédentes ont pour valeur le prix de la location de `lgv` pour le client représenté par le compte `cmt`.

Ecrivez ces deux méthodes. Assurez vous qu'elles continueront à fonctionner si le logiciel est étendu par de nouveaux sous-types de `Produit` et/ou de `Compte`. Expliquer pourquoi c'est le cas dans votre solution.

## 1.2 Locations, version 2

Le gérant du magasin décide de varier son offre et de créer des comptes avec forfaits. Les forfaits sont potentiellement cumulables et peuvent être à tout moment ajoutés à ou supprimés d'un compte existant. Voici des exemples de forfaits<sup>8</sup> :

- le forfait fidélité permet de gagner 10% de réduction supplémentaire chaque année.
- le forfait 20% permet d'obtenir pendant un an une réduction de 20% sur toute location,
- le forfait 50% coûte 30 euros à l'achat et permet d'obtenir pendant un an une réduction de 50% sur toute location.

### Questions.

1. Inspirez vous du schéma de conception "Decorateur" que vous avez étudié pour proposer une solution logicielle adaptée au problème posé ici, que vous décrirez par un texte et illustrerez par un diagramme de classes UML. N'oubliez pas d'indiquer la signature des méthodes utiles à l'exemple.
2. Dans le cadre de cette solution, donnez le code d'une classe définissant un compte avec forfait et de sa méthode `float prixFinalLocation(float prixCourant)`.
3. Quels limites voyez vous à ce schéma de conception ? Donner un exemple de forfait incompatible avec un autre.

---

7. Resp. le nom que vous lui avez donné à la question 1.

8. Bien sûr tous les forfaits ont un coût que nous ignorons dans cet énoncé.