

M1 Informatique (Master IFPRU)  
**Examen - FMIN107 ( “Ingénierie Logicielle” )**

Christophe Dony - Marianne Huchard - Clémentine Nebut

Session 1, année 2011-2012, janvier 2012.

Durée prévue : 2h00. Documents de cours et TD autorisés. Livres ou documents externes interdits. Les exemples de code de l'énoncé sont en Java, mais vous pouvez utiliser tout autre langage de programmation de votre choix pour vos réponses.

## Contexte

Un industriel vend des ordinateurs qu'il assemble lui-même à partir de composants <sup>1</sup>. Il permet également à ses clients de composer leurs propres ordinateurs à partir des composants existants. Il dispose d'une application informatique permettant de réaliser un assemblage et de calculer son prix Toutes Taxes Comprises ("All Taxes Included"), dit *prixTTC*. Le sujet porte sur cette application informatique réalisée avec le formalisme "objet". Elle vise au statut de framework en ce sens qu'elle est conçue pour être extensible et elle met en oeuvre différents schémas de conception (*Design Patterns*). Les exemples de l'énoncé sont donnés en Java mais lorsque du code est demandé, vous pouvez l'écrire dans le langage à objet de votre choix.

Dans cette application, on distingue les **composants simples** et les **montages**. Les **composant simples** sont de type "rack" (ou "tour", nom donné à l'armature externe), carte-mère, processeur, barrettes mémoire vive (RAM), disque-dur, lecteur DVD, carte vidéo, ventilateur, écran. Une fois l'assemblage en train de se faire ou réalisé, on voit apparaître des **montages** : carte-mère montée, ordinateur, réseau d'ordinateurs. Une carte-mère montée est composée d'une carte-mère, d'un processeur, d'un ventilateur, de RAM, etc. Un ordinateur peut être composé d'un rack, d'une carte-mère montée, d'un disque-dur, d'un écran, etc. L'ordinateur est un montage et également un composant car il peut être assemblé avec d'autres dans une configuration réseau. Il est possible d'ajouter un nombre quelconque de composants à un montage.

Le *prixTTC* d'un composant (simple ou montage) est le produit du prix hors taxe par la TVA. Le prix hors taxes (dit *prixHT*) des composants simples résulte d'un calcul sur la base du prix d'achat et de la marge bénéficiaire du vendeur. Le prix hors taxes d'un "montage" est la somme des prix hors taxes de ses composants.

## Architecture de Base (environ 8 points)

1. Un ordinateur, et plus généralement, tout montage est clairement représenté dans cette application par une arborescence d'objets qui relève du schéma de conception *Composite*. Nommons **ComposantOrdi** la racine de la hiérarchie de classes de notre mise en oeuvre. Elle contient une unique méthode non abstraite : `prixTTC()`. Son code est donné ci-dessous. Les méthodes `calculPrixHT()` et `getTVA()` sont abstraites sur cette classe.

```
public float prixTTC(){  
    return this.calculPrixHT() * this.getTVA();  
}
```

Donnez un diagramme de classes UML respectant le cahier des charges précédent. Placez y au moins une classe représentative des composants simples et une représentative des montages.

2. Du point de vue des schémas de réutilisation, comment la méthode `prixTTC()` de **ComposantOrdi** est-elle adaptable (ou paramétrable) ? Préciser comment, lors de son invocation, une affectation polymorphique est réalisée.

---

<sup>1</sup>Dans cet exercice, le terme "composant" ne dénote rien d'autre que dans le langage courant ; on dit par exemple qu'une carte video est l'un des composants d'un ordinateur.

3. Ecrivez la contrainte OCL précisant qu'un "montage" doit contenir au moins deux composants
4. Donnez une définition *Java* complète de la classe **Montage** qui représente les *montages*.
5. La classe **Montage** dispose d'une opération pour ajouter un composant (passé en argument) à un montage.
  - Ecrivez en OCL une précondition de cette opération consistant à vérifier que l'argument n'est pas déjà inclus dans le montage receveur, à n'importe quel niveau de son arborescence.
  - Ecrivez en OCL la postcondition de cette opération.
6. Que faut-il ajouter au code pour que l'on puisse écrire :

```
new CarteMereMontée()
    .addComponent(new CarteMere())
    .addComponent(new Processeur("Intel-i5"))
```

### Variation 1 avec le Schéma “Singleton” (environ 4 points)

7. Pour ne pas risquer de rupture de stock, chaque classe de composant simple doit être doté d'un mécanisme qui empêche de créer plus d'instances de la classe qu'il n'y a de produits réels en stock. En pensant au schéma *Singleton*, donnez la partie du code de la classe **DisqueDur** réalisant cette fonctionnalité.
8. En quoi cette implantation peut-elle différer selon que l'on code la solution en Java Standard ou en Smalltalk Standard ? (Un point de plus pour le code de la méthode appropriée de **DisqueDur** en Smalltalk).
9. Est-il possible de factoriser cette fonctionnalité sur la classe représentant les composants simples afin que chaque sorte ? Si oui dites comment (ne donnez pas de code).

### Variation 2 avec le schéma “Décorateur” (environ 4 points)

Suite aux évolutions du marché, certains barrettes mémoire sont soldées à la volée. Le concepteur de l'application souhaite doter la classe **RAM** d'un mécanisme de décoration, selon le schéma de conception *Décorateur* afin que l'on puisse ajouter ou retirer dynamiquement une décoration, représentant une réduction, à une instance de la classe **RAM**.

10. Donnez un texte décrivant la mise en oeuvre que vous proposez (respectant le schéma *Décorateur*).
11. Donnez un diagramme d'instance UML représentant une instance de la classe **RAM-Avec-Réduction** et le(s) autre(s) objet(s) qu'elle référence.
12. Donnez le code de la ou des méthode(s) `calculPrixHT()` que vous serez amenés à définir en disant sur quelle classe vous définissez chacune d'elles.

### Variation 3 avec le schéma “Bridge” (environ 2 points)

13. Pour des raisons de présentation la classe **DVD** est sous-classée par marques (nommons les *A* et *B*) mais pour chaque marque, les prix dépendent de la technologie utilisée (standard ou *blu-ray*). Proposez un découplage entre hiérarchies de marques et de technologies en vous inspirant du schéma de conception *Bridge* présenté en cours. Proposez un scénario réaliste pour le calcul du prix hors taxes dans un tel contexte.

### Autre Variation (environ 2 points)

14. Proposez une variation ou un problème lié à cet énoncé qui suggérerait l'utilisation d'un autre schéma de conception que vous connaissez.