Ml Informatique - HMIN102 - Ingenierie Logicielle

Annee 2015-2016, session 1de janvier 2016.

Duree : 2h00. Documents non autorises. La precision et la concision des reponses sont notees ainsi que la lisibilite des codes. Soyez precis et concis, vous gagnez du temps et faites plaisir aux correcteurs. Les exemple de code de l'enonce sont en Java, mais vous pouvez utiliser tout autre langage de programmation de votre choix

pour vos reponses. Tout schema UML doit comporter !'identification des relations, leur cardinalite, le nom des attributs et/ou methodes cle. Il est possible d'associer un code a une methode dans un schema de type UML,

comment clans le livre sur les design-patterns.

## Contexte pretexte

Un industriel vend des ordinateurs qu'il assemble lui-meme a partir de composants 1 Il dispose d'une application informatique permettant de realiser un assemblage et de calculer son prix Toutes Taxes Comprises ("All Taxes Included"), <lit *prixTTC.* Elle vise au statut de framework en ce sens qu'elle est conc;ue pour etre extensible ; elle met en oeuvre differents schemas de conception *( Design Patterns).*

.

On clistingue les composants simples et les montages. Les composant simples sont de type "rack" (ou ''tour", nom donne a !'armature externe), carte-mere, processeur, barrettes memoire vive (RAM), disque-dur,

lecteur DVD, carte video, ventilateur, ecran. Une fois l'assemblage en train de se faire ou realise, on voit apparaitre des montages : carte-mere montee, ordinateur, reseau d'ordinateurs. Une carte-mere montee est composee d'une carte-mere, d'un processeur, d'un ventilateur, de RAM, etc. Un ordinateur peut etre compose d'un rack, d'une carte-mere montee, d'un disque-dur, d'un ecran, etc. L'ordinateur est un montage et egalement

un composant car il peut etre assemble avec d'autres dans une configuration reseau. Il est possible d'ajouter un nombre quelconque de composants a un montage.

Le prix'ITC d'un composant (simple ou montage) est le produit du prix hors taxe par la TVA (taxe). Le prix hors taxes (<lit *prixHT)* des composants simples resulte d'un calcul sur la base du prix d'achat et de la marge beneficiaire du vendeur. Le prix hors taxes d'un "montage" est la somme des prix hors taxes de ses composants. La TVA peut varier selon les produits.

Dans l'application, les composants sont en toute generalites des ComposantsOrdi ; La classe ComposantOrdi detient une unique methode publique : prixTTC(). On utilise cette methode pour calculer le prixTTC de tout composantSimple ou Montage. On pose comme contrainte que cette methode ne soit definie qu'une seule fois

et n'ait jamais a etre modifiee OU redefinie.

## 1 Hierarchies et Architect ures extensibles (environ 6 points)

1. Pour la methode prixTTC() de ComposantOrdi, on choisit le parametrage par specialisation. Donnez le code de la classe *ComposantOrdi.*
2. Donnez un diagramme de classes UML representant les ComposantOrdi respectant le cahier des charges precedent. Placez y au moins une classe representative des composants simples et une representative des montages. Comentez l'architecture proposee.
3. Expliquez en quoi vous avez utilise le parametrage par specialisation.

Donnez le code de la methode prixHT() de votre classe representative des composants simples.

1. Donnez une definition complete de la classe Montage conforme a votre schema UML.
2. La description du schema *Composite* stipule que : *"When dealing with tree-structured data, programmers often have to discriminate between a leaf-node and a branch* ...".

Commentez cette phrase dans le contexte des deux questions precedentes.

1. Votre code permet-il d'ecrire le code suivant ? Si non modifiez le pour qu'il le permette.

neY CarteMereMontee ()

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

.add(new CarteMere ())

.add(new Processeur ("Intel-i5") )

# 2 Reutilisation en Typage Statique - environ 6 points

Dans une version Java du programme, on ajoute une methode de signature boolean equiv (ComposantOrdi c , String critere) sur la classe ComposantOrdi. Cette methode booleenne doit dire si un composant est

equivalent a un autre selon le critere donne.

Comme le calcul de l'equivalence entre deux montages est plus complexe et necessite un code specifique, on ajoute une methode de signature boolean equiv (Montage c , String critere sur la classe Montage pour ecrire le code traitant ce cas particulier.

Soient alors les 3 affectations et 8 envois de messages (appels de methodes) suivants 2 :

1 ComposantDrdi m1 = new RAM(); Montage m2 = new Montage(); ComposantOrdi m3 = new Montage();

s m2.equiv(m1 ,"x"); m2.equiv(m2 ,"x"); m2.equiv(m3,"x"); m2.equiv((Montage)m3,"x");

5 m3.equiv(m1 /'x"); m3.equiv(m2,"x"); m3.equiv(m3,"x"); m3.equiv((Montage)m3,"x");

1. Qualifiez et discutez !'instruction "ComposantOrdi m3 = new Montage () ;" dans le contexte de la reutilisation.
2. Pour chacun des 8 envois de messages precedents, indiquez, en prenant en compte les concepts de liaison dynamique, surcharge et/ou redefinition, laquelle des deux methodes equiv, celle de ComposantDrdi ou celle de Montage, est invoquee, et pourquoi.
3. Voyez vous, pour ces 8 cas, un ou plusieurs cas ou la methode Equiv de Montage n'est pas invoquee alors que ce serait necessaire ?
4. Si oui a la question precedente, indiquez ce que vous changeriez dans le programme pour qu'elle le soit.
5. Discutez clans le contexte de la question precedente et en lien avec la solution que vous aurez choisie, du concept de *downcasting,* et de celui de *double dispatch.*

# Variation sur l'architecture (1) (environ 5 points)

Suite aux evolutions du marche, certains types de memoire (RAM) de barettes memoires peuvent etre soldes et le rester pendant une duree de temps inconnue a l'avance.

* 1. Discutez de l'interet respectif des schemas *State* et *Decorator* pour integrer ces fonctionnalite de fan modulaire et extensible clans l'application et indiquez celle que vous utiliseriez.
  2. Pour votre solution preferee, donnez les elements des du code de votre solution. Les classes des, les envois de messages des qui font que cela fonctionne et que le resultat est extensible et modulaire doivent etre identifies. Entre autres, donnez le code de la (ou des) methode(s) prixHT().

# Variation sur I'architecture (2) (environ 3 points)

* 1. La constitution de montages coherents est clairement un probleme, par exemple on ne peut pas associer toute carte mere avec tout processeur. Par ailleurs, la selection un par un des composants peut etre fastidieuse, des montages pre-conc;us sont interessants. Discuter ces points en utilisant des schemas de conception et eventuellement morceaux de code interessants associes.
  2. L'argument "x" simule le critere de comparaison, ilest mis pour la coherence de l'enonce et n'a aucune importance dans les questions de ces exercices.