M1 Informatique - HMIN-102 - Ingénierie Logicielle
TD-TP
Autour du double dispatch et du schéma "Decorateur"

1 Locations de produits - version 1 - Utilisation du schéma "double-dispatch"

Considérons la modélisation objet d'un logiciel utilisé dans un magasin de location. Le magasin permet à ses abonnés d'acheter ou louer des produits. Pour louer, tout client doit posséder un compte. Le protocole pour simuler la location est le suivant (Listing 1) :

```
public static void main(String[] args){
    Produit lgv = new Produit("La grande vadrouille", 10.0);
    Client cl = new Client("Dupont");
    Compte cmt = new Compte(cl);
    cmt.prixLocation(lgv);
```

QUESTIONS

1. Réalisez un système informatique constitué des trois classes précédentes et d'un ensemble de méthodes avec les contraintes qu'il permette d'exécuter les instructions données au Listing 1 et qu'il soit extensible :
   — avec divers types de produits ayant des contraintes spécifiques quant aux prix de vente ou de location de base, par exemple on imagine une classe de produits soldés dont le prix de location est la moitié du prix de location d'un produit normal.
   — avec divers types de comptes, on pense par exemple à
   — une classe `CompteAvecReduction` offrant une réduction de 10% (par exemple) sur chaque location.
   — une classe `CompteAvecSeuil` offrant une location gratuite toutes les x locations.
   Aide : mettez en oeuvre une adaptation par spécialisation et trouvez une solution pour que, à chaque calcul du prix d'une location d'un produit pour un client, le produit, quel qu'il soit, et le compte, quel qu'il soit, collaborent (chacun des deux doit pouvoir dire son mot) pour calculer le prix de location final. Le système doit supporter l'ajout a posteriori de nouvelles sous-classes et les affectations polymorphiques éventuelles.
2. Donnez un code sans détails inutiles de ces deux classes de base et de leurs méthodes clé permettant d'exécuter correctement le listing 1 précédent.
3. Donnez le code de base des trois sous-classes `ProduitSoldé`, `CompteAvecReduction` et `CompteAvecSeuil` permettant d'exécuter le listing 2 ci-dessous. Si les classes de base de la question précédentes ont été bien réalisées, il ne sera pas nécessaire de les modifier. Pour argumenter votre le réalisation, étudiez le concept de "double dispatch" (voir liens internet) et établissez le rapport entre ce concept et votre solution.

```
    ...
    Compte cmt2 = new CompteAvecReduction(cl);
    System.out.println("CompteReduction : " + cmt2.prixLocation(lgv));

    Compte cmt3 = new CompteAvecSeuil(cl); // le seuil est à 2 par défaut
    System.out.println("CompteSeuil : " + cmt3.prixLocation(lgv));
    System.out.println("CompteSeuil : " + cmt3.prixLocation(lgv));
    System.out.println("CompteSeuil : " + cmt3.prixLocation(lgv)); // doit afficher 0

    Produit r4 = new ProduitSoldé("RockyIV", 10.0);
    System.out.println("CompteNormal+ProduitSoldé : " + cmt.prixLocation(r4));
    System.out.println("CompteReduction+ProduitSoldé : " + cmt2.prixLocation(r4));}
```

4. Est-il possible de réutiliser les classes précédentes pour réaliser des comptes avec réduction et avec seuil ?

2 Location de produits, version 2, utilisation du schéma "Décorateur"

QUESTIONS

1. Pour pallier la difficulté mise en évidence dans la dernière question de la section 1, réaliser le même cahier des charges global en utilisant le *pattern Decorateur*. Il permettra de décorer un compte avec différents forfaits dont le forfait *réduction* et le forfait *seuil* et surtout permettra de combiner les deux.
2. Quelles limites voyez-vous à ce schéma de conception ? Imaginez un exemple de forfait incompatible avec un autre.