



Licence 3

INF331 : Approche Orientée Objet de Modélisation des SI, UML
TD3 : Mise en œuvre des principes OO en Java

2023-2024

Valéry MONTHE

Exercice 1 : Classe simple et objet en Java

1. Créer une classe **Point** pour manipuler les points du plan. Un point a des coordonnées (abscisse, ordonnée). Cette classe doit disposer des trois méthodes suivantes :
 - **initialiser**: pour attribuer des valeurs aux coordonnées d'un point;
 - **deplacer**: pour modifier les coordonnées d'un point;
 - **afficher**: pour afficher un point ; par souci de simplicité, nous nous contenterons ici d'afficher les coordonnées du point (cette méthode affichera : «je suis un point de coordonnées X et Y»).
2. Ecrire un programme de test

Exercice 2 : Héritage

1. Créer une classe **Pointcol** pour manipuler les points colorés du plan. Elle doit avoir un attribut **couleur** de type byte et une opération **colorer**.
2. Modifier la classe **Pointcol**, en y ajoutant une méthode **afficheCol**, qui en plus des coordonnées du point coloré, affiche sa couleur.
3. Modifier la classe **Point**, y ajouter un constructeur. Puis ajouter également un constructeur dans la classe **Pointcol**, qui permet de construire les objets de point coloré, directement avec leur couleur.
4. Ecrire un programme de test

Exercice 3 : Polymorphisme

1. Modifier la classe **Pointcol**, pour que sa méthode **affichecol** s'appelle **afficher**.
2. Ecrire un exemple de programme qui exploite les possibilités de polymorphisme pour créer un tableau "hétérogène" d'objets, c'est-à-dire dans lequel les éléments peuvent être de type différent (**Point**, **Pointcol**). Puis le programme parcourt le tableau et affiche ses éléments (les points et les points colorés).

Exercice 4 : Polymorphisme

1. Modifier les classes **Point** et **Pointcol**, pour ne garder la méthode **afficher** que dans la classe **Point**.
2. Définir une méthode **identifie()** qui affiche pour :
 1. Un point : « je suis un point »
 2. Un Pointcol : « Je suis un point coloré de couleur "**couleur**" »
3. Ecrire un programme de test utilisant les possibilités de polymorphisme pour créer un tableau "hétérogène" d'objets.

Exercice 5 : Classe Abstraite

1. Créer une classe abstraite **Affichable**, ayant juste la signature d'une méthode **affiche** ().
2. Dériver deux classes, qui construisent respectivement les entiers et les flottants et les affichent :
 - a. Je suis un entier de valeur 25
 - b. Je suis un flottant de valeur 1.25

NB : Utiliser un tableau d'objets de type différents

3. Ecrire un programme de test

Exercice 6 : Interface

1. Transformer la classe abstraite **Affichable** de l'exercice précédent en interface.
2. Modifier les classes Entier et Flottant, pour obtenir le même résultat qu'à l'exercice précédent.
3. Ecrire un programme de test

Exercice 7 :

On veut gérer les comptes client dans une banque. On a modélisé un compte comme le montre le schéma ci-dessous.

Compte
-Numero : Chaine -Proprietaire : Chaine -Solde : Entier
+Deposer (montant : entier) +Retirer (montant : entier)

1. Ecrire le code de la classe Compte.
2. Créer une classe compte d'épargne (CompteEpargne) qui permet d'augmenter le solde suivant un certain taux d'intérêt.
3. On souhaite maintenant sécuriser la méthode retirer () de manière à n'autoriser un retrait uniquement que si le solde est suffisant. Créer une autre classe compte sécurisé (CompteSecurise) qui le fait.
4. Ecrire un programme de test