Université Abdelmalek Essaâdi
École Normale Supérieure-Tétouan

LP LDW
Année 2023-2024
Programmation JAVA

TD N°5

Exercice 1 :

On dispose de la classe suivante :

```
class Point
{
    public Point (double x, double y) { this.x=x ; this.y=y ; }
    public void deplace (double dx, double dy) { x+=dx ; y+=dy ; }
    public void affiche ()
    {
      System.out.println ("Point de coordonnees " + x + " " + y) ;
    }
    public double getX() { return x ; }
    public double getY() { return y ; }
    private double x, y ;
}
```

On souhaite réaliser une classe Cercle disposant des méthodes suivantes :

• **constructeur** recevant en argument les coordonnées du centre du cercle et son rayon,

• **deplaceCentre** pour modifier les coordonnées du centre du cercle,

• **changeRayon** pour modifier le rayon du cercle,

• **getCentre** qui fournit en résultat un objet de type Point correspondant au centre du cercle,

• **affiche** qui affiche les coordonnées du centre du cercle et son rayon.

1. Définir la classe Cercle comme classe dérivée de Point.

2. Définir la classe Cercle comme possédant un membre de type Point.

Dans les deux cas, on écrira un petit programme mettant en jeu les différentes fonctionnalités de la classe Cercle.

Exercice 2 :

Écrire une classe utilitaire ***UtilTab*** disposant des méthodes statiques suivantes :

• ***somme*** qui fournit la somme des valeurs d'un tableau de réels (*double*) de taille quelconque,

• ***incre*** qui incrémente d'une valeur donnée toutes les valeurs d'un tableau de réels (*double*).

Écrire un petit programme d'essai. Pour faciliter les choses, on pourra également doter la classe ***UtilTab*** d'une méthode d'affichage des valeurs d'un tableau de réels.

Exercice 3 :

Écrire une classe utilitaire **UtilTab** disposant des méthodes statiques suivantes :

• **genere** qui fournit en retour un tableau des n premiers nombres impairs, la valeur de n étant fournie en argument

• **somme** qui reçoit en argument deux vecteurs d'entiers de même taille et qui fournit en retour un tableau représentant la somme de ces deux vecteurs.

Écrire un petit programme d'essai. Pour faciliter les choses, on pourra également doter la classe **UtilTab** d'une méthode d'affichage des valeurs d'un tableau de réels.

Exercice 4 :

Ecrire 3 classes Points1, Points2 et Points3, permettant de créer des objets points dans les espaces dimension 1, dimension 2 et dimension 3. Chacune de ces classes doit être dotée d'au moins un constructeur.
Ecrivez toutes les méthodes que vous croyez pouvoir être appliquées à de tels objets.
Ensuite, créer un petit programme qui utilise ces points : affiche le nom de ces points, leur position, déplacement

Exercice 5 :

Ecrire trois classes : **Figure**, **Carre**, et **Rectangle**, telles que :

1. **Figure** a des attributs abscisse et ordonnée, ainsi qu'une couleur (encodée par un entier).
2. **Carre** et **Rectangle** héritent de **Figure**, mais ont en plus une ou deux longueurs pour les côtés.
3. **Figure** a un attribut privé **Vector** référençant toutes les instances de sa classe et de ses sous classes.
4. **Figure** a une méthode statique **getInstances()** renvoyant ce vecteur.
5. **Carre** et **Rectangle** redéfinissent cette méthode **getInstances()** de manière à ne récupérer que les instances qui correspondent à leur type.

Exercice 6 :

- **Un compte bancaire** possède à tout moment une donnée : son solde. Ce solde peut être positif (compte créditeur) ou négatif (compte débiteur).
- Chaque compte est caractérisé par un code incrémenté automatiquement.
- le code et le solde d'un compte sont accessibles en lecture seulement.
- A sa création, un compte bancaire a un solde nul et un code incrémenté.

- Il est aussi possible de créer un compte en précisant son solde initial.
- Utiliser son compte consiste à pouvoir y faire des dépôts et des retraits. Pour ces deux opérations, il faut connaître le montant de l'opération.
- L'utilisateur peut aussi consulter le solde de son compte par la méthode **ToString().**
- **Un compte Epargne** est un compte bancaire qui possède en plus un champ « Taux Intérêt=6 » et une méthode **calculIntérêt()** qui permet de mettre à jour le solde en tenant compte des intérêts.
- **Un ComptePayant** est un compte bancaire pour lequel chaque opération de retrait et de versement est payante et vaut 5 DH.

Questions :

1. Définir la classe Compte.
2. Définir la classe CompteEpargne.
3. Définir la classe ComptePayant.
4. Créer un programme permettant de tester ces classes avec les actions suivantes :
   - Créer une instance de la **classe Compte**, une autre de la **classe CompteEpargne** et une instance de la **classe ComptePayant**
   - Faire appel à la méthode **déposer()** de chaque instance pour déposer une somme quelconque dans ces comptes.
   - Faire appel à la méthode **retirer()** de chaque instance pour retirer une somme quelconque de ces comptes.
   - Faire appel à la méthode **calculInterêt()** du compte **Epargne**.
   - Afficher le solde des 3 comptes.