

TP8**Enseignants : Majdoub Manel, Madhi Kamel****Groupes : Prepa A02****Exercice 1**

1. Définissez les deux interfaces Moveable et Drawable comme suit :

```
public interface Drawable {  
    // choisis une couleur void setColor( int color);  
    // retourner la couleur int getColor();  
    // afficher void affiche(); }  
public interface Moveable { void translate( int dx , int dy ); }
```

2. Soit la classe Point définie comme suit :

```
public class Point implements Moveable {  
    private int x , y;  
    // Ajoutez 2 constructeurs : un constructeur par défaut  
    // et un autre de la forme Point ( int , int )  
    // Ajoutez des setters et des getters  
    // Ajouter la méthode toString()  
    //Ajouter les méthodes nécessaires }
```

3. Soit la classe Segment définie comme suit

```
public class Segment implements Drawable , Moveable {  
    Point endpoint1 ; Point endpoint2 ; int couleur ;  
    //Ajouter un constructeur Segment( Point , Point , int)  
    //Ajouter des setters et des getters  
    // Ajouter la méthode toString()  
    //Ajouter les méthodes nécessaires }
```

4. Soit la classe Triangle définie comme suit :

```
public class Triangle implements Drawable , Moveable {  
    private Segment segment1 ; private Segment segment2 ; private Segment seg-  
    ment3 ; private int couleur ;  
    //Ajouter un constructeur Triangle( Point , Point , Point , int )  
    //Ajouter des setters et des getters  
    // Ajouter la méthode toString()  
    //Ajouter les méthodes nécessaires }
```

5. Tester l'implémentation en écrivant une classe `FormeTest`.

Créer les objets suivants : un `Segment` et un `triangle`.

Traduire ces objets.

Créer un `ArrayList` comportant ces objets.

Afficher ces objets.

Exercice 2

On désire réaliser une application de gestion de comptes bancaires. Pour cela on a proposé d'organiser les objets de l'application en deux catégories (`CompteSurCarnet` et `CompteSurCheque`) et de les ranger dans une collection.

Les classes `CompteSurCarnet` et `CompteSurCheque` ont des propriétés communes telles que `Numéro` et `NomProprietaire` et `Solde` qui donnent naissance à une troisième classe `Compte` considérée comme classe de base pour éviter les redondances.

La classe `CompteSurCarnet` possède, en plus des propriétés communes, la propriété `NuméroCarnet`.

La classe `CompteSurCheque` est caractérisée par le `NuméroChèque` et le `NuméroCarte`.

1. Définir la classe abstraite `AbstractCompte` avec deux constructeurs (par défaut et d'initialisation) et les accesseurs.
 - Ajouter deux méthodes abstraites `créditer` et `Débit` qui prend en paramètre la somme à ajouter au compte ou à retirer du compte.
2. Définir la classe `CompteSurCheque` avec deux constructeurs (par défaut et d'initialisation) et les accesseurs.
 - Redéfinir la méthode `créditer` qui prend en paramètre la somme à ajouter au compte.
 - Redéfinir la méthode `Débit` qui prend en paramètre la somme à déduire du compte qui ne doit pas dépasser le solde sinon une exception est générée avec le message : « Solde insuffisant ».
3. Définir la classe `CompteSurCarnet` avec deux constructeurs (par défaut et d'initialisation) et les accesseurs.
 - Redéfinir la méthode `créditer` qui prend en paramètre la somme à ajouter au solde.
 - Redéfinir la méthode `Débit` qui prend en paramètre la somme à retirer du compte qui ne doit pas dépasser le solde sinon une exception est générée avec le message : « Solde insuffisant », en plus la somme à retirer ne doit pas dépasser un plafond de 10000D sinon une exception est générée avec le message : « Plafond dépassé ».

4. Définir la classe Banque contenant une collection (ArrayList) de comptes (CompteSurCheque et CompteSurCarnet) et ajouter les méthodes suivantes :
 - Rechercher : qui prend en paramètre un numéro et retourne l'indice du compte dans la collection ou -1 s'il n'existe pas.
 - AjouterCompteCarnet : qui prend les paramètres nécessaires à la création d'un compte sur carnet et de l'ajouter à la collection.
 - AjouterCompteCheque : qui prend les paramètres nécessaires à la création d'un compte sur chèque et de l'ajouter à la collection.
 - Supprimer : qui prend en paramètre un numéro et supprime le compte correspondant.
5. Définir la classe test contenant une fonction main permettant de tester les différentes classes.