Programmation orientée objet Java

TP1 : classes et objets

Exercice 1.

Réaliser une classe `Point` permettant de représenter un point sur un axe. Chaque point sera caractérisé par un nom (de type char) et une abscisse (de type double). On créera :

- Un constructeur recevant en arguments le nom du point et son abscisse.
- Une méthode affiche imprimant (en fenêtre de console) le nom du point et son abscisse.
- Une méthode translate effectuant une translation définie par la valeur de son argument.

Écrire un petit programme, dans une classe `TestPoint`, utilisant la classe `Point` pour créer un point, en afficher les caractéristiques, le déplacer et en afficher à nouveau les caractéristiques.

Exercice 2.

On dispose de la classe `Point` suivante permettant de manipuler des points d'un plan.

```
class Point
{
private double x, y ;
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 ("coordonnees = " + x + " " + y ) ; }

}
```

En ajoutant les fonctionnalités nécessaires à la classe `Point`, réaliser une classe `Segment` permettant de manipuler des segments d'un plan et disposant des méthodes suivantes :

```
Segment (Point or, Point ext)
Segment (double xOr, double yOr, double xExt, double yExt)
double longueur()
void deplaceOrigine (double dx, double dy)
void deplaceExtremite (double dx, double dy)
void affiche ()
```

Créer ensuite une classe `TestSegment` pour tester les fonctionnalités de la classe Segment.

Exercice 3.

On veut développer une mini-application de gestion des étudiants. Pour un étudiant, on retient comme informations : ID (entier auto-incrémenté), matricule (chaîne de caractères), prénom (chaîne de caractères), nom (chaîne de caractères), département (chaîne de caractères). Un élément de module a un code (chaîne de caractères), un titre (chaîne de caractères), un semestre d'étude (entier de 1 à 6), crédit (entier) et coefficient (réel).

1. Créer les classes Etudiant et ElementModule en ajoutant les méthodes utiles.
2. Créer une classe Note qui a trois attributs : etudiant (de type Etudiant), elementModule (de type ElementModule), moyenne (réel).
3. Créer une classe Util permettant de tester les classes précédentes.

    a. Créer un tableau de notes b.
    b. Afficher la note d'un étudiant donné pour élément identifié par un code.
    c. Afficher le nombre d'étudiants ayant validé un élément de module donné.
    d. Afficher la moyenne générale d'un étudiant identifié par un ID ou un matricule

4. On veut maintenant représenter un département par une classe ayant les attributs suivants : code (chaîne de caractères), nom (chaîne de caractères), nom responsable (chaîne de caractères).

1. Modifier la classe Etudiant pour prendre en considération ce changement.
2. Créer un constructeur de copie pour la classe Etudiant.