

TD 2 : Introduction à l'orienté objet
“Object-oriented design is the roman numerals of computing”

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 prévoira :

- Un constructeur recevant en arguments le nom et l'abscisse d'un point,
- Une méthode affiche imprimant (en fenêtre 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 utilisant cette classe pour créer un point, en afficher les caractéristiques, le déplacer et en afficher à nouveau les caractéristiques.

Exercice 2

Définir une classe Point caractérisée par son abscisse et ses coordonnées.
Définir :

- Le constructeur par défaut et d'initialisation de la classe.
- A l'aide des getters et les setters les méthodes d'accès aux attributs de la classe
- La méthode Norme () qui retourne la distance entre l'origine du repère et le point en cours

Écrire un programme permettant de tester la classe

Exercice 3

Ecrivez une classe Livre avec les attributs suivants:

- titre: Le titre du livre,
- auteur: L'auteur du livre,
- prix: Le prix du livre,
- annee: L'année du livre.

La classe Livre doit disposer des constructeurs suivants:

- Livre(),

- Livre(titre),
- Livre(titre, auteur),
- Livre(titre, auteur, prix),
- Livre(titre, auteur, prix, annee),
- Livre(Livre).

La classe Livre doit contenir des accesseurs et mutateurs pour les différents attributs. Elle doit aussi contenir une méthode toString() donnant une représentation de la classe Livre. Ecrivez aussi une classe de Main afin de tester la classe Livre.

Exercice 4

Ecrivez une classe Temps avec les attributs suivants:

- heures: De 0 à 23,
- minutes: De 0 à 59,
- secondes: De 0 à 59.

La classe Temps doit disposer des constructeurs suivants:

- Temps(): constructeur par défaut,
- Temps(heures),
- Temps(heures, minutes),
- Temps(heures, minutes, secondes),
- Temps(Temps).

La classe Temps doit contenir des accesseurs et mutateurs pour les différents attributs. Elle doit aussi contenir les méthodes:

- ajouterHeures(heures),
- ajouterMinutes(minutes),
- ajouterSecondes(secondes),
- toString() donnant une représentation de la classe Temps.

Ecrivez aussi une classe de Main afin de tester la classe Temps.

Exercice 5

Ecrivez une classe Rectangle avec les attributs suivants:

- longueur,
- largeur.

La classe Rectangle doit disposer des constructeurs suivants:

- Rectangle(): constructeur par défaut,
- Rectangle(longueur, largeur),
- Rectangle(Rectangle).

La classe Rectangle doit contenir des accesseurs et mutateurs (qui vérifient que les valeurs sont positives) pour les différents attributs. Elle doit aussi contenir les méthodes:

- perimetre: Le périmètre est égal à $2 * (longueur + largeur)$,
- aire: L'aire est égale à $longueur * largeur$,
- isCarre: vérifie si le rectangle est un carré,
- toString: donne une représentation d'un rectangle.

Ecrivez aussi une classe testRectangle afin de tester la classe Rectangle.

Exercice 3

Banque		
-	NCompte : int	
-	Solde : float	
-	CIN : String	
+	<<Constructor>> Banque (int NCompte, float Solde, String CIN)	
+	deposer (float Somme)	: void
+	retirer (float Somme)	: void
+	avoirSolde ()	: float
+	avoirInf ()	: String

1. Ecrire le programme permettant d'implémenter la classe "Banque"
2. Ecrire un programme Main pour :
 - Créer un compte: 1, 5000.75, "AB 1200"
 - Afficher les informations de ce compte
 - Afficher son solde
 - Déposer 500
 - Afficher les informations de ce compte
 - Afficher son solde
 - Retirer 200
 - Afficher les informations de ce compte
 - Afficher son solde