

# TP N° 1

## Avant de commencer :

Conversion de types :

`Integer.parseInt (String c)` : convertir la chaîne c en un entier

`Integer.toString(int i)` : convertir l'entier i en une chaîne de caractères

`Float.parseFloat(String c)` : convertir la chaîne c en un réel

`Float.toString(float f)` : convertir le réel f en une chaîne de caractères

De même pour double, long, byte,...

## Exercice 1

Ecrire une classe *Rectangle* qui contient :

- deux attributs privés entiers x et y représentant respectivement la longueur et la largeur du rectangle,
- un constructeur permettant d'initialiser x et y avec deux valeurs différentes passées en arguments,
- une méthode nommée *perimetre ()* qui retourne le périmètre du rectangle
- une méthode nommée *surface ()* qui retourne la surface du rectangle.

Ecrire une classe publique nommée *ExerciceRectangle* contenant la méthode *main* qui prend en ligne de commande la longueur et la largeur d'un rectangle et affiche le périmètre et la surface de celui-ci.

## Exercice 2 :

- 1) Définir une classe nommée *Vecteur* avec les attributs privés x et y (de type entier) et les méthodes suivantes :
  - « *afficher ()* » : affiche les composantes du vecteur,
  - « *produitScalaire ()* » : calcule le produit scalaire de deux vecteurs et retourne le résultat.
- 2) Ajouter 2 constructeurs :
  - le premier constructeur permet d'initialiser x et y avec la même valeur passée en argument,
  - le deuxième constructeur permet d'initialiser x et y avec deux valeurs différentes passées en argument.
- 3) Ajouter une classe publique *TestVecteur* contenant la méthode *main ()* permettant d'initialiser deux vecteurs v1 et v2 (les valeurs sont passées en ligne de commande), d'afficher l'abscisse et l'ordonnée de chacun, de calculer leur produit scalaire et de l'afficher.

## Exercice 3 :

Soit un ensemble d'étudiants.

- 1) Définir une classe nommée *Etudiant* qui se caractérise par un nom (chaîne de caractères privée) et 4 notes (double privé).
- 2) Ajouter un constructeur qui initialise les différents attributs de la classe,
- 3) Ecrire une méthode protégée qui affiche les notes de chaque étudiant,

- 4) Ecrire une méthode publique qui calcule et retourne la moyenne de l'étudiant,
- 5) Ecrire une méthode qui affiche si l'étudiant est « admis » ou « non admis ».
- 6) Ajouter une classe publique *TestEtudiant* contenant la méthode main permettant de calculer la moyenne et d'afficher le résultat (admis ou non admis) pour les deux étudiants suivants :
  - Ali, 11, 13, 18, 7
  - Faten, 15, 9, 8, 16

#### **Exercice 4**

- 1) Réaliser une classe *Point* permettant de représenter un point sur un axe. Chaque point sera caractérisé par un nom (de type *char privé*) et une abscisse (de type *double privé*). On prévoira :
  - a) un constructeur recevant en arguments le nom et l'abscisse d'un point,
  - b) une méthode publique nommée *affiche* imprimant (en fenêtre console) le nom du point et son abscisse,
  - c) une méthode protégée nommée *translate* effectuant une translation définie par la valeur de son argument.
- 2) Ajouter une classe publique *TestPoint* contenant la méthode main et permettant de :
  - a) créer un point avec les valeurs suivantes 'A', 12.5,
  - b) en afficher les caractéristiques,
  - c) le déplacer de 1.5
  - d) en afficher à nouveau les caractéristiques.

#### **Exercice 5**

Ecrire une classe *Médicament* contenant :

- 1) Trois attributs privés :
  - ✓ **Nom** : nom du *médicament*.
  - ✓ **Prix** : le prix du médicament.
  - ✓ **Stock** : la quantité stockée du médicament.
- 2) Les constructeurs :
  - ✓ Un constructeur permettant de créer un médicament à partir du son nom.
  - ✓ Un constructeur permettant de créer un médicament à partir du son nom et du prix.
  - ✓ Un constructeur permettant de créer un médicament à partir du son nom et du stock.
  - ✓ Un constructeur permettant de créer un médicament à partir du son nom, son prix et son stock.
- 3) Une méthode nommée *augmenterStock()* recevant en paramètre la quantité en stock et permettant d'augmenter le stock.
- 4) Une méthode nommée *diminuerStock()* recevant en paramètre la quantité en stock et permettant de diminuer le stock.
- 5) Les méthodes : *getStock()*, *getPrix()*, *getNom()* permettant de récupérer le nom, le prix et le stock.
- 6) Une méthode nommée *affiche()* permettant d'afficher les informations d'un médicament sous la forme suivante : Le médicament #Nom# a un prix de : #prix# et un stock de : #stock#.
- 7) Définir des méthodes permettant de modifier le contenu des attributs caractérisant la classe.

- 8) Ajouter une classe publique *TestMedicament* contenant la méthode main permettant de :
- Créer les médicaments suivants:
    - ✓ Fervex : stock : 20, prix : 5DT
    - ✓ Aspirine : 1500 DT
    - ✓ Adol.
  - Ajouter la quantité 40 pour aspirine. Afficher le résultat de l'ajout.
  - Ajouter quantité 30 pour Fervex. Afficher le résultat obtenu.
  - Modifier le prix du Aspirine à 1200. Afficher le prix modifié.