

Travaux dirigés n° 1

Introduction à la POO (1)

Exercice 1 (Les vecteurs)

Un vecteur est caractérisé par un nom et deux valeurs : une abscisse et une ordonnée

1°) Définissez les attributs (noms et types) d'un objet **Vecteur** et réalisez le diagramme de classe correspondant.

2°) Écrivez la classe **Vecteur** avec ses attributs ainsi que le constructeur par initialisation.

3°) Accès aux attributs et modification :

- Écrivez les *getters* pour chaque attribut.
- Quel(s) attribut(s) est-il pertinent de pouvoir modifier ?
- En fonction de votre réponse, proposez le/les *setter(s)* pour la classe **Vecteur**.

4°) La classe **TestVecteur** :

a) Écrivez une classe **TestVecteur** contenant un **main** et permettant d'utiliser la classe **Vecteur**. Les informations sont saisies par l'utilisateur au clavier. Puis vous afficherez les caractéristiques du vecteur saisi.

b) Donnez l'état mémoire au cours de l'exécution du **main**.

Exercice 2 (Les livres)

Un livre est caractérisé par un identifiant (le numéro ISBN, que nous considérerons comme un entier), un titre, et un prix (et beaucoup d'autres choses que nous négligeons ici). Nous désirons écrire une classe **Livre** dont les objets sont des livres au sens précédent.

1°) Définissez les attributs (noms et types) d'un objet **Livre**. Écrivez-les, ainsi que les constructeurs par défaut et par initialisation.

2°) Accès aux attributs et modification :

- Écrivez les *getters* pour chaque attribut.
- Quel(s) attribut(s) est-il pertinent de pouvoir modifier ?
- En fonction de votre réponse, proposez le/les *setter(s)* pour la classe **Livre**.

3°) Écrivez une méthode **toString** qui construit et retourne une chaîne de caractères contenant le titre du livre et son prix.

4°) Donnez la représentation (le diagramme) UML de la classe **Livre**.

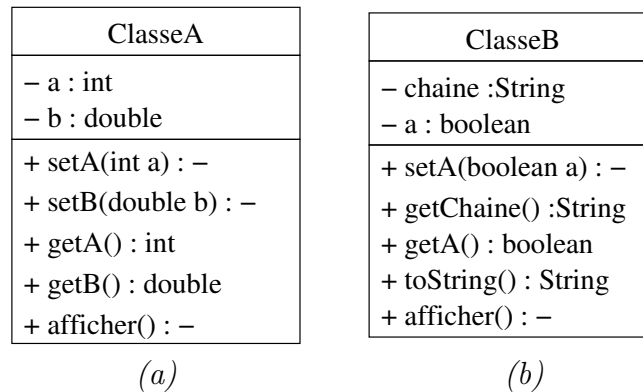
5°) La classe **TestLivre** :

a) Écrivez une classe **TestLivre** contenant un **main** et permettant d'utiliser la classe **Livre**. Les informations sont saisies par l'utilisateur au clavier.

b) Donnez l'état mémoire au cours de l'exécution du **main**.

Exercice 3 (Du diagramme UML à la classe)

Donnez le code correspondant aux classes représentées par les diagrammes UML ci-dessous. Vous considérerez un constructeur par défaut et par initialisation pour chaque classe.



Exercice 4 (Les cercles)

Un cercle est caractérisé par deux coordonnées et un rayon (trois réels).

1°) Conception, réalisation

- Donnez le code de la classe **Cercle** : attributs, getters et setters.
- Précisez ce qu'est, pour vous, un cercle par défaut.
- Complétez le code de la classe **Cercle** pour qu'elle dispose de deux constructeurs : constructeur par défaut et constructeur par initialisation.
- Comment peut-on réagir si on tente de construire un cercle avec une valeur négative du rayon ? Si ce n'est déjà fait, modifiez le constructeur par initialisation pour prendre en charge cette "bonne réaction".
- Complétez le code de la classe **Cercle** par une méthode **toString** pour obtenir les caractéristiques

2°) Donnez le diagramme UML permettant de représenter la classe **Cercle**.

3°) Écrivez le code Java d'une classe de test permettant :

- de créer le cercle unité ;
- de créer un autre cercle, dont le centre est de coordonnées 3.5 et -1.6 et dont le rayon est 1
- de calculer la distance entre l'origine et le centre du second cercle.
- d'afficher les caractéristiques des 2 cercles uniquement si l'abscisse du centre est strictement positive.

4°) Donnez le diagramme UML permettant de décrire cette version de la classe **Cercle**.