

Programmation en Java

Applications 02 suite : POO en Java

Pour mener à bien cet atelier, chaque étudiant devra créer un projet par exercice en suivant la notation :

Appli01_ExoX_Prenom_Nom ou *Appli01_ExoX_Groupe_NumeroDuGroupe*

Le *X* doit être remplacé le numéro correspondant de l'exercice. *Prenom* et *Nom* doit être remplacé par votre prenom et nom sans les accents. *NumeroDuGroupe* doit être remplacé par le numéro du Groupe.

Exercice 1 :

- Définir une classe **Livre** avec les attributs suivants : Titre, Auteur (Nom complet), Prix.
- Définir à l'aide des propriétés les méthodes d'accès aux différents attributs de la classe.
- Définir un **constructeur** permettant d'initialiser les attributs de la méthode par des valeurs saisies par l'utilisateur.
- Définir la méthode **Afficher()** permettant d'afficher les informations du livre en cours.
- Écrire un programme testant la classe **Livre**.

| Exemple d'exécution :

```
Livre 1:
Donner le titre: Programmer en C
Donner l'auteur: Claude Delannoy
Donner le prix 350
Le titre est Programmer en C
L'auteur est Claude Delannoy
Le prix est 350
Titre: Programmer en C, Auteur: Claude Delannoy ,Prix: 350
Livre 2:
Donner le titre: Programmer en Java
Donner l'auteur: Claude Delannoy
Donner le prix 450
Titre: Programmer en Java, Auteur: Claude Delannoy ,Prix: 450
```

Exercice 2 :

- Définir une classe Rectangle ayant les attributs suivants : Longueur et Largeur.
- Définir à l'aide des propriétés les méthodes d'accès aux attributs de la classe.
- Ajouter un constructeur d'initialisation.
- Ajouter les méthodes suivantes :
 - Périmètre() : retourne le périmètre du rectangle.
 - Aire() : retourne l'aire du rectangle.
 - EstCarre() : vérifie si le rectangle est un carré.
 - AfficherRectangle() : expose les caractéristiques d'un rectangle comme suit :
Longueur : [...] - Largeur : [...] - Périmètre : [...] - Aire : [...] - Il s'agit d'un carré / Il ne s'agit pas d'un carré

| Exemple d'exécution :

Rectangle 1

Donner la longueur du rectangle: 3

Donner la largeur du rectangle: 4

Le périmètre est 14

La surface est 12

Ce n'est pas un carré

Rectangle 2

Donner la longueur du rectangle: 5

Donner la largeur du rectangle: 5

- Longueur : 5- Largeur : 5- Périmètre : 20- Aire : 25- Il s'agit d'un carré

Exercice 3 :

1. Écrire une classe Complexes permettant de représenter des nombres complexes. Un nombre complexe Z comporte une partie réelle et une partie imaginaire : $Z = \text{PartieRéelle} + \text{PartieImaginaire} * i$
2. Définir à l'aide des propriétés les méthodes d'accès aux attributs de la classe.
3. Définir un **constructeur** par défaut permettant d'initialiser les deux parties du nombre à 0.

4. Définir un **constructeur** d'initialisation pour la classe.
5. Ajouter les méthodes suivantes :
 - **Plus(Complexe)** : Elle permet de retourner le nombre complexe obtenu en ajoutant au nombre en cours un nombre complexe passé en argument.
 - **Moins(Complexe)** : Elle permet de retourner le nombre complexe obtenu en soustrayant au nombre en cours un nombre complexe passé en argument.
 - **Afficher ()** : Elle donne une représentation d'un nombre complexe comme suit : $a+bi$.
6. Écrire un programme permettant de tester la classe Complexe.

Exemple d'exécution :

```

Nombre Complexe 1:
Donner la partie réelle: 2
Donner la partie imaginaire: 4
Nombre Complexe 1: 2+4i
Nombre Complexe 2:
Donner la partie réelle: 3
Donner la partie imaginaire: -2
Nombre Complexe 2: 3-2i

La somme:
5+2i

La Différence:
-1+6i

```

Exercice 4 :

1. Définir une classe Employé caractérisée par les attributs : **Matricule**, **Nom**, **Prénom**, **DateNaissance**, **DateEmbauche**, **Salaire**.
2. Définir à l'aide des propriétés les méthodes d'accès aux différents attributs de la classe.
3. Définir un constructeur permettant d'initialiser les attributs de la méthode par des valeurs saisies par l'utilisateur.
4. Ajouter à la classe la méthode **Age()** qui retourne l'âge de l'employé.
5. Ajouter à la classe la méthode **Anciennete()** qui retourne le nombre d'années d'ancienneté de l'employé.

6. Ajouter à la classe la méthode **AugmentationDuSalaire()** qui augmente le salaire de l'employé en prenant en considération l'ancienneté.

Si Ancienneté < 5 ans, alors on ajoute 2%. - Si Ancienneté < 10 ans, alors on ajoute 5%. - Sinon, on ajoute 10%.

7. Ajouter la méthode **AfficherEmployé()** qui affiche les informations de l'employé comme suit :

- **Matricule** : [...]

- **Nom complet** : [NOM Prénom]

- **Age** : [...]

- **Ancienneté** : [...]

- **Salaire** : [...]

Le nom doit être affiché en majuscule. Pour le prénom, la première lettre doit être en majuscule, les autres en minuscule.

8. Ecrire un programme de test pour la classe Employé.

Exemple d'exécution :

```
Matricule: 10941
Nom: SALIM
Prénom: omar
Date de naissance (jj/mm/aaaa): 04/08/1990
Date embauche (jj/mm/aaaa): 5/11/2012
salaire: 10000

Matricule : 10941
Nom Complet : SALIM Omar
Age : 25
Ancienneté : 3
Salaire : 10000

Après application de l'augmentation de salaire

Matricule : 10941
Nom Complet : SALIM Omar
Age : 25
Ancienneté : 3
Salaire : 10200
```

Exercice 5 :

1. Créer la classe Article caractérisée par les attributs : Référence, Désignation, PrixHT, TauxTVA.

Ces attributs doivent seulement être accessibles par le biais des accesseurs (get / set) en lecture/écriture mis en œuvre par les propriétés.

2. Ajouter les constructeurs suivants :

- Un constructeur par défaut
- Un constructeur initialisant tous les attributs.
- Un Constructeur qui permet de renseigner la référence et la désignation lors de l'instanciation
- Un constructeur de recopie

3. Implémentez la méthode CalculerPrixTTC() ;

Cette méthode doit calculer le prix TTC d'un article qui équivaut à : $\text{PrixHT} + (\text{PrixHT} * \text{TauxTVA} / 100)$ et retournera la valeur calculée.

4. Ajouter la méthode **AfficherArticle()** qui affiche les informations de l'article.
5. Créer un programme de test où il faut créer des objets (en utilisant les différents constructeurs) et leur calculer le prix TTC.
6. Le taux de TVA est en fait commun à tous les articles. Pour éviter toute redondance de cet attribut, vous devriez donc la déclarer comme partagée au niveau de la classe Article et non comme un attribut spécifique des objets instanciés à partir de la classe. Proposer une solution et tester de nouveau.

Exemple d'exécution :

```
Donner le taux de TVA pour tous les articles : 20  
Le taux TVA est : 20%
```

```
Article 1:  
Référence : 0  
Désignation :  
Prix HT : 0  
Le prix TTC est 0
```

```
Article 2:  
Donner la référence: 111  
Donner la désignation: ATA  
Donner le prix HT: 100  
Référence : 111  
Désignation : ATA  
Prix HT : 100  
Le prix TTC est 120
```

```
Article 3:  
Donner la référence: 122  
Donner la désignation: RER  
Référence : 122  
Désignation : RER  
Prix HT : 0  
Le prix TTC est 0
```

```
Article 4:  
Référence : 111  
Désignation : ATA  
Prix HT : 100  
Le prix TTC est 120
```