



# **JAVA POO**

---

**Concepteur Développeur D'Applications**

CDA24060

Août 2025

# Concept de base

- Définir les concepts de classes, objets, attributs, et méthodes.
- Rôle du constructeur.

# Exercice 1



Réaliser la classe correspondante

- 1 – Créer une classe Java nommée Compte qui représente un compte bancaire, ayant pour attributs : public Double solde
- 2 – Créer un constructeur ayant comme paramètre solde.
- 3 – Créer une méthode déposer() du type void qui gère les versements
- 4 – Créer une méthode retirer() du type void qui gère les retraits.
- 5 – Créer une méthode afficher() du type void permettant d'afficher le solde
- 7 – Créer une classe TestCompte permettant de tester le compte en effectuant un versement et puis un retrait

# Exercice 2



Réaliser la classe correspondante

1. Créer une classe Voiture Java de visibilité public, ayant pour attributs : String marque, Double prix,
2. Créer un constructeur sans paramètres ( par défaut )
3. Créer les getters et setters
4. Créer une méthode void afficher( ) permettant d'afficher les résultats
5. Créer une classe Exécution permettant d'exécuter les résultats.

# Exercice 3



Réaliser la classe correspondante

1. Créer une classe Java nommée Rectangle ayant pour attributs : Longueur et Largeur tous deux du type Double,
2. ajouter ensuite un constructeur avec paramètres.
3. Définir une méthode surface() du type Double permettant de calculer la surface du rectangle et une méthode périmètre() du type Double permettant de calculer le périmètre du rectangle.
4. Créer une méthode afficher() du type void pour afficher les résultats. Donner ensuite le code complet de la classe.
5. Créer une classe TestRectangle pour afficher les résultats surface et périmètre.

# Exercice 4



Réaliser les classes correspondantes

1. Créer un package Java nommé geometry
2. Au sein du package geometry créer une classe Java nommée Point ayant pour attribut Double Abscisse et double ordonnee
3. Au sein du même package créer une classe Cercle doté d'un attribut centre du type Point et d'un rayon de type Double.
4. Et des méthodes suivantes : périmètre() du type Double permettant de calculer le périmètre du cercle. surface() du type Double permettant de calculer la surface du cercle.
5. testAppartenance() du type void permettant de tester si un point appartient au cercle ou non
6. afficher() du type void permettant d'afficher les résultats
7. Créer une classe TestCercle contenant une méthode static void main() permettant de tester les résultats

# Encapsulation

- Utiliser l'encapsulation et les accesseurs.
- Comprendre le rôle du mot clés public et private

# Exercice 1

Voici une classe Livre.java

- Corrigez les erreurs de syntaxe
- Ajoutez une méthode main() pour créer 2 livres.
- Faire afficher les auteurs de ces 2 livres.

```
public class Livre {  
    // Variables  
    private String titre, auteur;  
    private int nbPages  
  
    // Constructeur  
    public Livre(String unAuteur, String unTitre) {  
        auteur = unAuteur;  
        titre = unTitre;  
    }  
  
    // Accesseur  
    public String getAuteur() {  
        return auteur;  
    }  
  
    // Modificateur  
    void setNbPages(int n) {  
        nbPages = nb;  
    }  
}
```



# Exercice 2



## Accesseurs et modificateurs

1. Modifiez la classe Livre :
  1. Ajoutez un accesseur pour la variable titre et la variable nbPages.
  2. Ajoutez un modificateur pour les variables auteur et titre.
  3. Changez le modificateur de nbPages : il ne devra changer le nombre de pages que si on lui passe en paramètre un nombre positif, et ne rien faire sinon, en affichant un message d'erreur.
2. Dans la méthode main(),
  1. indiquez le nombre de pages de chacun des 2 livres,
  2. faites afficher ces nombres de pages,
  3. calculez le nombre de pages total de ces 2 livres et affichez-le.

# Exercice 3

---

Suite

1. Dans La classe Livre, ajoutez une méthode afficheToi() qui affiche une description du livre (auteur, titre et nombre de pages).
2. Ajoutez 2 constructeurs pour avoir 3 constructeurs dans la classe :
  1. Un constructeur qui n'a pas de paramètre
  2. Un qui prend en paramètre l'auteur et le titre du livre,
  3. et l'autre qui prend en plus le nombre de pages.
  4. Utilisez les 3 constructeurs (et éventuellement d'autres méthodes) pour créer 3 livres de 300 pages dans la méthode main() de la classe TestLivre

# Exercice 4

Contrôle des variables private par les modificateurs

1. Contrôle des variables private par les modificateurs
  1. Ajoutez un prix aux livres (nombre qui peut avoir 2 décimales ; type Java float ou double) avec 2 méthodes getPrix et setPrix pour obtenir le prix et le modifier.
  2. Ajoutez au moins un constructeur qui prend le prix en paramètre.
  3. On bloque complètement les prix : un prix ne peut être saisi qu'une seule fois et ne peut être modifié ensuite (une tentative pour changer le prix ne fait qu'afficher un message d'erreur). Réécrivez la méthode setPrix (et autre chose si besoin est). Vous ajouterez une variable booléenne prixFixe (pour "prix fixé") pour savoir si le prix a déjà été fixé.
  4. Réécrire la méthode main () et prévoir le deux cas ( prix non fixé ou bien prix fixé plusieurs fois ) afficher le résultat de l'exécution.