

Série 3 POO partie 1

Exercice 1 : Création d'objets et Encapsulation

Créez une classe `Voiture` avec les attributs suivants :

- `marque` (String)
- `modele` (String)
- `annee` (int)

Rendez ces attributs **privés** et créez des **getters et setters** pour les manipuler.

Dans la méthode `main`, créez un objet `Voiture`, définissez ses attributs via les setters et affichez-les avec les getters.

Exercice 2 : Constructeurs et Surcharge

Créez une classe `Livre` avec les attributs :

- `titre` (String)
- `auteur` (String)
- `anneePublication` (int)

Ajoutez **trois constructeurs** :

1. Un constructeur avec tous les paramètres.
2. Un constructeur avec seulement `titre` et `auteur`, et `anneePublication` fixé à 2020 par défaut.
3. Un constructeur sans paramètre, assignant "Inconnu" à `titre` et `auteur`, et 2000 à `anneePublication`.

Dans `main`, testez ces constructeurs en créant plusieurs objets.

Exercice 3 : Getters/Setters et Vérification des Données

Créez une classe `CompteBancaire` avec les attributs :

- `numeroCompte` (String)
- `solde` (double)

Ajoutez des **getters et setters** et assurez-vous que :

- Le solde ne peut jamais être négatif (dans le setter).
- Ajoutez une méthode `deposer(double montant)` qui ajoute de l'argent.
- Ajoutez une méthode `retirer(double montant)` qui enlève de l'argent **seulement si le solde est suffisant**.

Dans `main`, créez un compte, essayez de déposer et retirer de l'argent et testez les restrictions.

Exercice 4 : Encapsulation et Sécurité des Données

Créez une classe `Utilisateur` avec :

- `identifiant` (`String`)
- `motDePasse` (`String`)

Mettez les attributs en `private` et :

1. Ajoutez un setter pour `motDePasse`, qui ne modifie la valeur **que si elle fait au moins 8 caractères**.
2. Ajoutez une méthode `afficherInfos()` qui ne retourne **pas** le mot de passe mais affiche `"Identifiant: [identifiant]"`.

Dans `main`, testez la classe en créant un utilisateur et en essayant de lui attribuer un mot de passe trop court.

Exercice 5 : Surcharge de Méthodes

Créez une classe `Rectangle` avec les attributs :

- `longueur` (`double`)
- `largeur` (`double`)

Ajoutez :

1. Une méthode `calculerSurface()` qui retourne `longueur * largeur`.
2. Une surcharge `calculerSurface(double facteur)` qui applique un coefficient multiplicateur à la surface.

Dans `main`, créez un rectangle et testez les différentes versions de `calculerSurface()`.