**Travaux Dirigés N° 9**

**Encapsulation et abstraction en Java.**

**Niveau Débutant**

**Exercice 1 : Classe Personne**

Créez une classe Personne :

* Attributs privés : nom (String) et age (int).
* Fournissez un constructeur pour initialiser les valeurs.
* Ajoutez des **getters** et **setters** pour lire et modifier nom et age.
* Ajoutez une méthode publique afficherDetails() pour afficher les informations.

Testez la classe dans une méthode main.

**Exercice 2 : Classe Produit**

Créez une classe Produit :

* Attributs privés : nom (String), prix (double) et quantiteStock (int).
* Ajoutez des méthodes pour :
  + Modifier la quantité en stock (ajouter ou retirer des unités).
  + Calculer la valeur totale en stock (prix \* quantiteStock).
* Testez les fonctionnalités avec différents produits.

**Exercice 3 : Classe abstraite Animal**

Créez une classe abstraite Animal :

* Attributs : nom (String) et age (int).
* Méthode abstraite parler() : chaque animal parle différemment.
* Méthode concrète manger() : affiche "L'animal mange".

Implémentez une sous-classe Chien où la méthode parler() affiche "Le chien aboie".

**Niveau Intermédiaire**

**Exercice 4 : Classe CompteBancaire**

Créez une classe CompteBancaire :

* Attributs privés : numeroCompte (String) et solde (double).
* Méthodes publiques :
  + deposer(double montant) : augmente le solde (si montant > 0).
  + retirer(double montant) : réduit le solde (si montant ≤ solde).
  + getSolde() : retourne le solde.
* Implémentez une règle pour empêcher les dépôts ou retraits invalides et affichez un message d’erreur.

Testez plusieurs scénarios dans une méthode main.

**Exercice 5 : Classe abstraite Forme**

Créez une classe abstraite Forme :

* Méthode abstraite calculerAire() : calcule l'aire.
* Méthode concrète afficherAire() : affiche l'aire.

Créez deux sous-classes :

* Rectangle : calcule l'aire à partir de la longueur et la largeur.
* Cercle : calcule l'aire à partir du rayon (utilisez Math.PI).

Testez la classe avec un rectangle et un cercle.

**Exercice 6 : Gestion des employés**

Créez une classe Employe :

* Attributs privés : nom, id et salaire (double).
* Méthodes :
  + Constructeur pour initialiser les valeurs.
  + Getter pour afficher le salaire.
  + Méthode publique augmenterSalaire(double pourcentage) : augmente le salaire par un pourcentage donné.

Ajoutez une classe Main pour créer des employés, modifier leurs salaires et afficher les détails.

**Niveau Avancé**

**Exercice 7 : Gestion des véhicules**

Créez une classe abstraite Vehicule :

* Attributs : marque (String) et modele (String).
* Méthodes :
  + Constructeur pour initialiser les valeurs.
  + Méthode abstraite deplacer() : comportement spécifique selon le véhicule.
  + Méthode concrète afficherDetails() : affiche la marque et le modèle.

Implémentez deux sous-classes :

* Voiture : la méthode deplacer() affiche "La voiture roule sur la route".
* Bateau : la méthode deplacer() affiche "Le bateau navigue sur l'eau".

**Exercice 8 : Gestion des formes géométriques**

Créez une classe abstraite Forme :

* Attributs : couleur (String).
* Méthode abstraite calculerAire().

Implémentez trois sous-classes :

* Triangle : calcule l’aire à partir de la base et de la hauteur.
* Carre : calcule l’aire à partir d’un côté.
* Cercle : calcule l’aire à partir du rayon.

Dans une méthode main, créez une liste de formes et affichez l'aire de chaque forme.

**Exercice 9 : Gestion d'une bibliothèque**

Créez une classe abstraite Document :

* Attributs : titre et auteur (String).
* Méthodes :
  + Constructeur pour initialiser les valeurs.
  + Méthode abstraite afficherType() : affiche le type de document.

Créez deux sous-classes :

* Livre : ajoute l'attribut nbPages.
* Article : ajoute l'attribut nomMagazine.

Dans une méthode main, créez plusieurs documents, affichez leurs types et leurs détails.

**Exercice 10 : Système de gestion scolaire**

Créez une classe abstraite Personne :

* Attributs : nom (String) et id (int).
* Méthode abstraite afficherRole().

Implémentez deux sous-classes :

* Etudiant : ajoute un attribut niveau (ex. : "Licence 1").
* Professeur : ajoute un attribut salaire.

Dans une méthode main, créez une liste de personnes (étudiants et professeurs) et affichez leur rôle ainsi que leurs détails.