**Travaux Dirigés N° 10**

**🟢 Niveau Facile (Débutant)**

**Exercice 1 : Classe Animal (Héritage Simple)**

Créez une classe Animal avec :

* Attributs : nom (String).
* Méthodes : manger() qui affiche "L’animal mange.".

Créez une sous-classe Chat qui hérite de Animal et ajoute :

* Une méthode miauler() qui affiche "Le chat miaule.".

Dans main(), créez un objet Chat, assignez-lui un nom, appelez manger() et miauler().

**Exercice 2 : Surcharge de méthodes (Calculatrice)**

Créez une classe Calculatrice avec plusieurs méthodes additionner() :

1. additionner(int a, int b) qui retourne a + b.
2. additionner(double a, double b) qui retourne a + b.
3. additionner(int a, int b, int c) qui retourne a + b + c.

Dans main(), testez ces méthodes avec différents types de données.

**Exercice 3 : Héritage avec constructeur**

Créez une classe Vehicule avec :

* Attributs : marque (String), vitesse (int).
* Constructeur pour initialiser marque et vitesse.
* Méthode afficherInfos() qui affiche la marque et la vitesse.

Créez une sous-classe Voiture qui hérite de Vehicule et ajoute :

* Un attribut nombrePortes (int).
* Un constructeur qui initialise tous les attributs.
* Une méthode afficherDetails() qui affiche les informations de la voiture.

Dans main(), créez un objet Voiture et affichez ses détails.

**🟡 Niveau Intermédiaire**

Ces exercices introduisent la redéfinition des méthodes et l'utilisation du polymorphisme.

**Exercice 4 : Redéfinition de méthode (Employe)**

Créez une classe Employe avec :

* Attributs : nom (String), salaire (double).
* Constructeur et méthode afficherInfos() qui affiche le nom et le salaire.

Créez une sous-classe Manager qui hérite de Employe et redéfinit afficherInfos() pour inclure une **prime**.

Dans main(), créez un Employe et un Manager et appelez afficherInfos() sur les deux.

**Exercice 5 : Polymorphisme avec redéfinition (Instrument)**

Créez une classe Instrument avec :

* Une méthode jouer() qui affiche "L’instrument produit un son.".

Créez deux sous-classes Guitare et Piano qui redéfinissent jouer() :

* Guitare affiche "La guitare joue une mélodie.".
* Piano affiche "Le piano joue une symphonie.".

Dans main(), créez une **liste d’instruments** et appelez jouer() sur chaque objet.

**Exercice 6 : Héritage en cascade (Animal - Mammifere - Chien)**

Créez une hiérarchie avec trois niveaux :

1. Animal (méthode respirer()).
2. Mammifere qui hérite de Animal et ajoute marcher().
3. Chien qui hérite de Mammifere et ajoute aboyer().

Dans main(), créez un Chien et appelez toutes les méthodes disponibles.

**🔴 Niveau Difficile**

Ces exercices demandent une gestion avancée du polymorphisme et de la hiérarchie des classes.

**Exercice 7 : Gestion des véhicules (Polymorphisme)**

Créez une classe Vehicule avec une méthode deplacer().

Créez deux sous-classes :

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

Dans main(), créez une **liste de véhicules** et appelez deplacer() sur chaque objet.

**Exercice 8 : Classe abstraite Forme**

Créez une classe abstraite Forme avec :

* Une méthode abstraite calculerAire().

Créez deux sous-classes :

* Rectangle avec longueur et largeur, implémente calculerAire().
* Cercle avec rayon, implémente calculerAire().

Dans main(), créez une **liste de formes** et affichez l'aire de chaque.

**Exercice 9 : Gestion des employés avec polymorphisme**

Créez une classe Employe avec une méthode calculerSalaire().

Créez deux sous-classes :

* EmployePleinTemps : avec un salaire fixe.
* EmployeTempsPartiel : avec un taux horaire et heures travaillées.

Dans main(), créez des objets EmployePleinTemps et EmployeTempsPartiel et affichez leurs salaires.

**Exercice 10 : Bibliothèque de documents**

Créez une classe abstraite Document avec :

* Attributs : titre, auteur.
* Méthode abstraite afficherType().

Créez deux sous-classes :

* Livre qui ajoute nbPages.
* Article qui ajoute nomMagazine.

Dans main(), créez une liste de Document et affichez leurs types et détails.