

PARTIE 2 — Classes, objets, encapsulation

Exercice 1 : Créer une classe `Personne`

Objectif : Créer une classe avec nom, prénom, âge, et une méthode `sePresenter()` qui affiche une phrase.

Template :

```
public class Personne {
    // TODO : attributs privés

    // TODO : constructeur

    // TODO : méthode sePresenter()

    // TODO : getters/setters
}
```

Classe test :

```
public class TestPersonne {
    public static void main(String[] args) {
        Personne p = new Personne("Alice", "Durand", 30);
        p.sePresenter();
    }
}
```

Exercice 2 : Classe `CompteBancaire`

Objectif : Créer un compte avec solde, dépôt et retrait. Empêcher un retrait si le solde est insuffisant.

Template :

```
public class CompteBancaire {
    private double solde;

    public CompteBancaire(double soldeInitial) {
        this.solde = soldeInitial;
    }

    public void deposer(double montant) {
        // TODO
    }

    public void retirer(double montant) {
        // TODO
    }

    public double getSolde() {
        return solde;
    }
}
```

PARTIE 3 — Héritage, interfaces, polymorphisme

Exercice 1 : Héritage avec `Animal`

Objectif : Créer une classe `Animal` puis des classes filles `Chien` et `Chat`, chacune redéfinissant une méthode `crier()`.

Test polymorphe :

```
public class TestAnimaux {
    public static void main(String[] args) {
        Animal[] animaux = { new Chien(), new Chat() };

        for (Animal a : animaux) {
            a.crier(); // Appelle la bonne méthode selon l'objet réel
        }
    }
}
```

Exercice 2 : Classe abstraite `Employe`

Objectif : Créer une classe abstraite `Employe` avec une méthode abstraite `calculerSalaire()`. Implémenter des classes `Salarie` et `Freelance`.

Templates :

```
public abstract class Employe {
    protected String nom;

    public Employe(String nom) {
        this.nom = nom;
    }

    public abstract double calculerSalaire();
}

public class Salarie extends Employe {
    private double salaireMensuel;

    public Salarie(String nom, double salaire) {
        // TODO
    }

    @Override
    public double calculerSalaire() {
        return salaireMensuel;
    }
}

public class Freelance extends Employe {
    private int heures;
    private double tarifHoraire;

    public Freelance(String nom, int heures, double tarif) {
        // TODO
    }
}
```