

## Projet : Application de Gestion des Employés

### Contexte

Je suis chargée de développer une application pour gérer les employés d'une entreprise. Cette application doit utiliser les concepts de la programmation orientée objet, avec une structure claire et un code facilement maintenable. Les données des employés seront représentées par des objets, et je devrai manipuler un tableau d'objets pour gérer les employés.

### Objectifs de l'application

1. **Ajouter un employé** : Créer un nouvel objet `Employe` et l'ajouter au tableau.
2. **Modifier un employé** : Mettre à jour les informations d'un employé existant à l'aide de son identifiant unique.
3. **Supprimer un employé** : Retirer un employé du tableau grâce à son identifiant unique.
4. **Afficher la liste des employés** : Afficher tous les employés avec leurs détails.
5. **Rechercher un employé** : Trouver un employé par son nom ou son poste.
6. **Calculer la masse salariale** : Calculer la somme totale des salaires mensuels des employés.
7. **Trier les employés** : Afficher les employés triés par salaire (ordre croissant ou décroissant).

### Spécifications Techniques

#### 1. Création d'une classe `Employe`

##### *Attributs privés :*

- `int id` : Identifiant unique de l'employé.
- `String nom` : Nom complet de l'employé.
- `String poste` : Poste occupé par l'employé.
- `double salaire` : Salaire mensuel de l'employé.

##### *Constructeurs :*

- Un constructeur avec tous les attributs.

- Un constructeur par défaut.

#### **Méthodes :**

- Des **getters et setters** pour accéder et modifier les attributs.
- Une méthode `toString()` pour afficher les informations complètes d'un employé.
- Une méthode statique `compareParSalaire` pour comparer deux employés par leur salaire (utile pour le tri).

## **2. Tableau d'objets Employe**

- J'ai utilisé un tableau de type `Employe[]` pour stocker les employés (limité à 50 employés maximum).
- J'ai ajouté une vérification pour m'assurer que le tableau n'est pas plein avant d'ajouter un nouvel employé.

## **3. Classe principale GestionEmployes**

Cette classe contient les méthodes suivantes et le menu principal :

#### **Méthodes statiques principales :**

1. `printMenu()` : Afficher le menu principal avec les options disponibles.
2. `ajouterEmploye(Employe employe)` : Ajouter un nouvel objet `Employe` au tableau.
3. `modifierEmploye(int id, String nouveauNom, String nouveauPoste, double nouveauSalaire)` : Modifier un employé existant.
4. `supprimerEmploye(int id)` : Supprimer un employé du tableau à l'aide de son identifiant.
5. `afficherEmployes()` : Afficher tous les employés du tableau.
6. `rechercherEmploye(String critere)` : Rechercher un employé par nom ou poste, et afficher ses informations.
7. `calculerMasseSalariale()` : Calculer la somme des salaires des employés présents dans le tableau.
8. `trierEmployesParSalaire(boolean ordreCroissant)` : Trier et afficher les employés par salaire dans l'ordre choisi (croissant ou décroissant).

## Ecrans montrant l'exécution de chaque besoin

### Ajout d'un employé

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.1\lib\idea_rt.jar=5000:C:\Program Files\JetBrains\IntelliJ IDEA 2024.1\bin" -Dfile.encoding=UTF-8

--- Menu Gestion des Employés ---
1. Ajouter un employé
2. Modifier un employé
3. Supprimer un employé
4. Afficher la liste des employés
5. Rechercher un employé
6. Calculer la masse salariale
7. Trier les employés par salaire
8. Quitter
Votre choix : 1
ID: 1
Nom: EL HAITI IKRAME
Poste: dev
Salaire: 5000
Employé ajouté avec succès !
```

### Modification des informations d'un employé

```
--- Menu Gestion des Employés ---
1. Ajouter un employé
2. Modifier un employé
3. Supprimer un employé
4. Afficher la liste des employés
5. Rechercher un employé
6. Calculer la masse salariale
7. Trier les employés par salaire
8. Quitter
Votre choix : 2
ID de l'employé à modifier : 1
Nouveau nom : Ikrame
Nouveau poste : dev1
Nouveau salaire : 4500
Modification réussie !
```

## Suppression d'un employé

```
--- Menu Gestion des Employés ---
1. Ajouter un employé
2. Modifier un employé
3. Supprimer un employé
4. Afficher la liste des employés
5. Rechercher un employé
6. Calculer la masse salariale
7. Trier les employés par salaire
8. Quitter
Votre choix : 3
ID de l'employé à supprimer : 1
Employé supprimé avec succès !
```

## Affichage de la liste des employés

```
--- Menu Gestion des Employés ---
1. Ajouter un employé
2. Modifier un employé
3. Supprimer un employé
4. Afficher la liste des employés
5. Rechercher un employé
6. Calculer la masse salariale
7. Trier les employés par salaire
8. Quitter
Votre choix : 4
Employe{id=1, nom='ikrame', poste='dev', salaire=5000.0}
Employe{id=2, nom='omar', poste='dev', salaire=5000.0}
Employe{id=3, nom='naima', poste='tech', salaire=7000.0}
Employe{id=4, nom='hamou', poste='prof', salaire=8000.0}
Employe{id=5, nom='ayoub', poste='tech', salaire=9000.0}
```

## Recherche d'un employé

```
--- Menu Gestion des Employés ---
1. Ajouter un employé
2. Modifier un employé
3. Supprimer un employé
4. Afficher la liste des employés
5. Rechercher un employé
6. Calculer la masse salariale
7. Trier les employés par salaire
8. Quitter
Votre choix : 5
Saisie nom ou poste à rechercher : dev
Employe{id=1, nom='ikrame', poste='dev', salaire=5000.0}
Employe{id=2, nom='omar', poste='dev', salaire=5000.0}
```

```
--- Menu Gestion des Employés ---
1. Ajouter un employé
2. Modifier un employé
3. Supprimer un employé
4. Afficher la liste des employés
5. Rechercher un employé
6. Calculer la masse salariale
7. Trier les employés par salaire
8. Quitter
Votre choix : 5
Saisie nom ou poste à rechercher : ayoub
Employe{id=5, nom='ayoub', poste='tech', salaire=9000.0}
```

## Calcul de la masse salariale

```
--- Menu Gestion des Employés ---
1. Ajouter un employé
2. Modifier un employé
3. Supprimer un employé
4. Afficher la liste des employés
5. Rechercher un employé
6. Calculer la masse salariale
7. Trier les employés par salaire
8. Quitter
Votre choix : 6
Masse salariale totale : 34000.0
```

## Tri des employés par salaire

```
--- Menu Gestion des Employés ---
1. Ajouter un employé
2. Modifier un employé
3. Supprimer un employé
4. Afficher la liste des employés
5. Rechercher un employé
6. Calculer la masse salariale
7. Trier les employés par salaire
8. Quitter
Votre choix : 7
Ordre croissant (true) ou décroissant (false) ? true
Employe{id=1, nom='ikrame', poste='dev', salaire=5000.0}
Employe{id=2, nom='omar', poste='dev', salaire=5000.0}
Employe{id=3, nom='naima', poste='tech', salaire=7000.0}
Employe{id=4, nom='hamou', poste='prof', salaire=8000.0}
Employe{id=5, nom='ayoub', poste='tech', salaire=9000.0}
```

```
--- Menu Gestion des Employés ---
1. Ajouter un employé
2. Modifier un employé
3. Supprimer un employé
4. Afficher la liste des employés
5. Rechercher un employé
6. Calculer la masse salariale
7. Trier les employés par salaire
8. Quitter
Votre choix : 7
Ordre croissant (true) ou décroissant (false) ? false
Employe{id=5, nom='ayoub', poste='tech', salaire=9000.0}
Employe{id=4, nom='hamou', poste='prof', salaire=8000.0}
Employe{id=3, nom='naima', poste='tech', salaire=7000.0}
Employe{id=1, nom='ikrame', poste='dev', salaire=5000.0}
Employe{id=2, nom='omar', poste='dev', salaire=5000.0}
```

## Difficultés rencontrées

Lors du développement de cette application, j'ai rencontré certaines difficultés :

- **Gestion des erreurs utilisateur** : L'utilisateur pouvait entrer des valeurs incorrectes (ex: un salaire négatif). Pour résoudre cela, j'ai ajouté des vérifications.
- **Manipulation du tableau statique** : L'utilisation d'un tableau de taille fixe limitait la flexibilité de l'application. Une alternative serait d'utiliser une liste dynamique (ArrayList).
- **Tri des employés** : L'implémentation du tri a nécessité l'utilisation d'une lambda expression pour la comparaison des salaires.

## **Conclusion**

Ce projet m'a permis de mettre en pratique plusieurs concepts fondamentaux de la programmation en Java, notamment la gestion des objets et des tableaux. Des améliorations possibles incluent une interface graphique et l'utilisation d'une base de données pour stocker les employés de manière persistante.

**EL HAITI IKRAME**

**05/02/2025**

**Java**