

Rapport – Application de Gestion des Employés

1. Présentation de l'application

L'application Gestion des Employés est un programme Java en mode console permettant de gérer efficacement les informations des employés d'une entreprise.

Elle repose sur la programmation orientée objet (POO) avec une classe `Employe` représentant chaque employé, et une classe `GestionEmployes` assurant la gestion du tableau d'employés ainsi que l'interaction avec l'utilisateur.

FONCTIONNALITÉS PRINCIPALES :

1. Ajouter un employé avec identifiant, nom, poste et salaire.
2. Modifier un employé existant grâce à son identifiant.
3. Supprimer un employé du tableau.
4. Afficher tous les employés avec leurs détails.
5. Rechercher un employé par nom ou poste.
6. Calculer la masse salariale totale.
7. Trier les employés par salaire (ordre croissant ou décroissant).

2. Étapes de réalisation

ÉTAPE 1 – CONCEPTION DE LA CLASSE EMPLOYE :

- Création des attributs privés : id, nom, poste, salaire.
- Mise en place de constructeurs : par défaut et avec paramètres.
- Implémentation des getters et setters.
- Méthode `toString()` pour afficher les informations formatées.
- Méthode statique `compareParSalaire` pour comparer deux employés (utile pour le tri).

ÉTAPE 2 – CRÉATION DE LA CLASSE PRINCIPALE GESTIONEMPLOYES

- Déclaration d'un tableau `Employe[]` (max 50 éléments).
- Création des méthodes :
 - `ajouterEmploye`
 - `modifierEmploye`
 - `supprimerEmploye`
 - `afficherEmployes`

- rechercherEmploye
- calculerMasseSalariale
- trierEmployesParSalaire
- Mise en place du menu principal avec boucle while et switch.

ÉTAPE 3 – TESTS ET VALIDATION

- Lancement de l'application dans IntelliJ IDEA.
- Tests successifs de chaque fonctionnalité avec différents jeux de données.
- Vérification des cas limites (tableau vide, identifiant inexistant, etc.).

3. ÉCRANS D'EXÉCUTION

Menu principal

```
=== GESTION DES EMPLOYES ===  
1. Ajouter un employé  
2. Modifier un employé  
3. Supprimer un employé  
4. Afficher la liste des employés  
5. Rechercher un employé (par nom ou poste)  
6. Calculer la masse salariale  
7. Trier les employés par salaire  
0. Quitter  
Choix :
```

Ajout d'un employé

```
Choix : 1  
ID employé (entier) : 01  
Nom complet : WISSAL EZOUINE  
Poste : CADRE  
Salaire mensuel : 8000  
Employé ajouté.
```

Modification d'un employé

```
Choix : 2
ID de l'employé à modifier : 01
WISSAL EZOUINE
Nouveau nom (laisser vide pour conserver : WISSAL EZOUINE) : Nouveau poste (laisser vide pour conserver : CADRE) : DEVELOPPEUR IT
Nouveau salaire (>=0) : 15000
Employé modifié.
```

Suppression d'un employé

```
Choix : 3
ID de l'employé à supprimer : 02
Confirmer suppression (o/N) ? 0
Employé supprimé.
```

Recherche par nom ou poste

```
Choix : 5
Critère de recherche (nom ou poste) : DEVELOPPEUR IT
Résultats :
ID: 1 | Nom: WISSAL EZOUINE | Poste: DEVELOPPEUR IT | Salaire: 15000,00
```

Calcul de la masse salariale

```
Choix : 6
Masse salariale totale : 15000,00
```

Tri par salaire

```
Choix : 7
Tri croissant ? (o/N) : 0
Triage terminé (croissant).

--- Liste des employés (2) ---
ID: 1 | Nom: WISSAL EZOUINE | Poste: DEVELOPPEUR IT | Salaire: 15000,00
ID: 2 | Nom: TAHA KADA | Poste: CHEF DE PROJET | Salaire: 20000,00
```

4. Difficultés rencontrées

- **Gestion des entrées utilisateur : validation des types (entier pour ID, double pour salaire).**
- **Manipulation du tableau : suppression et décalage des éléments sans collections.**
- **Tri des employés : implémentation d'un algorithme de tri manuel.**
- **Prévention des doublons : s'assurer que deux employés n'ont pas le même identifiant.**

Fin du rapport
