



Assigné

FR

## Gestion des Employés

MS

Mly Youssef SBAI

22.12.2020

Gestions des Employés de TangerLab

MySQL

Java

Git

UML

### Livrables

Dépôt GitHub Diagramme de Classes

### Référentiels

- Développeur-se web et web mobile
- Concepteur-riceréveloppeur-se d'applications

### Contexte du projet

Le directeur de l'entreprise TangerLab spécialisée dans les produits chimiques souhaite gérer les salaires et primes de ses employés au moyen d'un programme Java. Un employé est caractérisé par son nom, son prénom, son âge et sa date d'entrée en service dans l'entreprise. Ecrivez une classe abstraite Employe dotée des attributs nécessaires, d'une méthode abstraite calculerSalaire (ce calcul dépendra en effet du type de l'employé) et d'une méthode getNom retournant une chaine de caractère obtenue en concaténant la chaine de caractères "L'employé " avec le prénom et le nom. Dotez également votre classe d'un constructeur prenant en paramètre l'ensemble des attributs nécessaires.

#### Calcul du salaire

Le calcul du salaire mensuel dépend du type de l'employé. On distingue les types d'employés suivants : Ceux affectés à la Vente. Leur salaire mensuel est le 20 % du chiffre d'affaire qu'ils réalisent mensuellement, plus 1500 Dh. Ceux affectés à la Représentation. Leur salaire mensuel est également le 20 % du chiffre d'affaire qu'ils réalisent mensuellement, plus 2500 Dh. Ceux affectés à la Production. Leur salaire vaut le nombre d'unités produites mensuellement multipliées par 5. Ceux affectés à la Manutention. Leur salaire vaut leur nombre d'heures de travail mensuel multipliées par 50 Dh. Codez dans votre fichier Salaires.java une hiérarchie de classes pour les employés en respectant les conditions suivantes : \*La super classe de la hiérarchie doit être la classe Employe. \* Les nouvelles classes doivent contenir les attributs qui leur sont spécifiques ainsi que le codage approprié des méthodes calculerSalaire et getNom, en changeant le mot "employé" par la catégorie correspondante. \* Chaque sous classe est dotée de constructeur prenant en argument l'ensemble des attributs nécessaires. \* N'hésitez pas à introduire des classes intermédiaires pour éviter au maximum les redondances d'attributs et de méthodes dans les sous classes

#### Employés à risques

Certains employés des secteurs production et manutention sont appelés à fabriquer et manipuler des produits dangereux. Après plusieurs négociations syndicales, ces derniers parviennent à obtenir une prime de risque mensuelle. Complétez votre programme Salaires.java en introduisant deux nouvelles sous classes d'employés. Ces sous classes désigneront les employés des secteurs production et manutention travaillant avec des produits dangereux. Ajouter également à votre programme une interface pour les employés à risque permettant de leur associer une prime mensuelle fixe de 200.

#### Collection d'employés

Satisfait de la hiérarchie proposée, notre directeur souhaite maintenant l'exploiter pour afficher le salaire de tous ses employés ainsi que le salaire moyen. Ajoutez une classe Personnel contenant une "collection" d'employés. Il s'agira d'une collection polymorphique d'Employe.

Définissez ensuite les méthodes suivantes à la classe Personnel : \* void ajouterEmploye(Employe): qui ajoute un employé à la collection. \* void calculerSalaires() : qui affiche le salaire de chacun des employés de la collection. \* double salaireMoyen(): qui affiche le salaire moyen des employés de la collection.

#### Modélisation

Réaliser le diagramme correspondant à cet exemple Base de données Créer une base de donnés Mysql nommée GestionPersonnel Ajouter toutes les tables nécessaires pour alimenter la base de données

#### Méthode Main

L'utilisateur Ajoute un Seul enregistrement pour chaque type d'employé L'utilisateur peut modifier ou supprimer un employé Confirmation des opérations (Ajout, Suppression, Modification) dans la base de donnés

### Modalités pédagogiques

Travail Individuel

Durée : 4 jours

### Compétences visées

- C5. Créer une base de données

niveau 1, imiter

niveau 2, adapter

niveau 3, transposer
- C6. Développer des composants d'accès aux données

niveau 1, imiter

niveau 2, adapter

niveau 3, transposer
- C7. Développer la partie back-end d'une application web ou web mobile

niveau 1, imiter

niveau 2, adapter

niveau 3, transposer

+ 8 compétences