

I-

Une entreprise souhaite gérer les salaires de ces employés. Pour cela, elle crée une classe

Employe ayant les caractéristiques suivantes :

- Une donnée membre char* nom
- Une donnée membre char* prenom
- Une donnée membre int age
- Une donnée membre int ancienneté (le nombre d'années de la personne dans l'entreprise)
- Une méthode double calculer_base_salaire() qui calcule la base du salaire de la personne.
- Une méthode void afficher() qui affiche la personne avec sa profession
- Une méthode double calculer_salaire() qui calcule le salaire de la personne (base de salaire plus 10% par année d'ancienneté)
- Un constructeur par défaut et une autre avec paramètres.
- Un destructeur
- Un constructeur par recopie
- Un opérateur d'affectation
- Surcharger les opérateurs >> et <<
- Surcharger l'opérateur ==

On souhaite pouvoir définir les classes dérivées d'Employé suivantes :

– Une classe **Commercial** ayant une donnée membre double chiffre_affaire et deux données statiques constantes double BASE et double PART. La base de salaire est donnée par la formule suivante :

$$\text{BASE} + \text{chiffre_affaire} * \text{PART}$$

– Une classe **Technicien** ayant deux données membres char* grade et int nb_unité_produite et trois données statiques constantes double BASE, double PART et double GAIN_UNITE. La base de salaire est donnée par la formule suivante :

$$\text{BASE} + \text{nb_unité_produite} * \text{PART} * \text{GAIN_UNITE}$$

– Une classe **Manutentionnaire** ayant une donnée membre int Nb_heures et une donnée statique constante double BASE_SALAIRE_HORAIRE. La base de salaire est donnée par la formule suivante :

$$\text{BASE_SALAIRE_HORAIRE} * \text{Nb_heures}.$$

1- Quelles sont les fonctions qui doivent être déclarées en tant que fonctions virtuelles dans la classe Employé ? Justifier pourquoi ?

2- Quelles sont les fonctions qui doivent être déclarées en tant que fonctions virtuelles pures dans la classe Employé ? Justifier ! La classe **Employe** est-elle une classe abstraite ?

3- Ajouter à chaque classe les constructeurs et les destructeurs correspondants.

4- Ajouter à chaque classe un constructeur par recopie et un opérateur d'affectation.

5- Surcharger les opérateurs >> et <<

6- Surcharger l'opérateur ==

7- Implémenter les différentes classes et tester-les.

II:

On souhaite créer de nouvelles classes **Vendeur** dérivée de Commercial et **Représentant** dérivée de Commercial et Manutentionnaire:

- **Vendeur** dont le salaire est calculé comme un Commercial
 - **Représentant** dont le salaire est calculé comme un Manutentionnaire
1. Que faut-il changer dans la classe Commercial et Manutentionnaire pour implémenter ces deux classes ?
 2. Définir les constructeurs et les destructeurs associés à ces deux classes.
 3. Définir un constructeur par copie pour chaque classe
 4. Définir un opérateur d'affectation pour chaque classe
 5. Surcharger les opérateurs >> et <<
 6. Surcharger l'opérateur ==
 7. Donner le diagramme de classe associé à ce projet.
 8. Ce diagramme de classe pose un problème, lequel ? Proposer la solution.
 9. Implémenter ces deux classes et tester-les.