

## PARTIE 1 :

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

**Employé** ayant les caractéristiques suivantes :

- Une donnée membre char\* nom
- Une donnée membre char\* prénom
- 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)
- Des constructeurs associés (par défaut et avec paramètres) et le constructeur par copie
- Un destructeur
- Un opérateur d'affectation

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 une donnée membre 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 ?
2. Quelles sont les fonctions qui doivent être déclarées en tant que fonctions virtuelles pures dans la classe Employé ? La classe employé est-elle une classe abstraite ?
3. Implémenter les différentes classes et tester-les.

## PARTIE 2:

On souhaite créer de nouvelles classes dérivées de Commercial :

- **Vendeur** dont le salaire est calculé comme un Commercial
- **Représentant** dont le salaire est calculé comme un Manutentionnaire

1. Que faut-il changer à la classe Commercial pour implémenter ces deux classes ?
2. Implémenter ces deux classes et tester-les.
3. Il-y-t-il de la duplication de code entre Manutentionnaire et Représentant ? Si oui,

comment y remédier ?

### **PARTIE 3:**

On souhaite créer de nouvelles classes dérivées d'Employé correspondant aux employés avec prime de risque (un montant différent pour chaque employé qui est rajouté à la base de salaire) :

**ManutentionnaireARisque et TechnicienARisque.**

1. Que faut-il changer aux classes Manutentionnaire et Technicien pour implémenter ces deux classes ?
2. Comment éviter la duplication de code pour la gestion des primes de risque dans ces deux classes ?
3. Implémenter ces deux classes et tester-les.

### **PARTIE 4:**

On souhaite créer une classe **Personnel** contenant :

- Un tableau d'Employés (sous forme d'un pointeur).
- Une méthode void embaucher(Employé\* newbie) qui ajoute un employé
- Une méthode void licencie(Employé\* e) qui supprime l'employé e
- Une méthode void licencie() qui supprime tous les employés
- Une méthode void afficher\_salaires() qui affiche le salaire de tous les employés
- Une méthode double salaire\_moyen() qui calcule le salaire moyen des employés
- Un constructeur
- Un destructeur

1. De quelles façons peut-on déterminer si un employé e est dans le vecteur extensible de Personnel ? Faut-il rajouter une méthode à la classe Employé ?
2. Implémenter cette classe et tester-la.