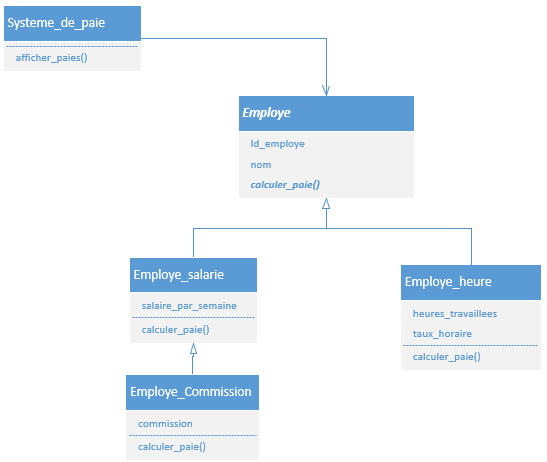
# **Exercice :** Méthode abstraite calculer\_paie() et ses implémentations dans les classes dérivées.

Cette flèche indique une RELATION non spécifique et NON une relation d’héritage

## Modélisation UML des objets :



## Mise en contexte :

Le système de paie veut calculer et afficher la paie des employés.

Mais on a des employés payés à la semaine pour un montant fixe.

On a aussi des employés payés à la semaine pour un montant fixe ET qui ont aussi une commission.

On a aussi des employés payés à l'heure

Tous les différents types d’employés héritent de la classe Employe. Nous allons faire en sorte que cette classe soit abstraite (ABC) et ait une méthode abstraite (en italique dans les diagrammes UML) ***calculer\_paie()***.

Cela obligera les classes dérivées à implémenter la méthode calculer\_paie() également.

## À partir du diagramme UML, faites la création des différentes classes dans Python.

Comme les calculs à faire pour les différentes méthodes calculer\_paie() sont très simples, vous n’avez pas besoin de pseudo-code ni de diagramme de flux pour les coder.

Vous pouvez voir qu’il y a de l’héritage.

Commencez par la classe Employe.

Puis faites les classes dérivées.

Quand on calcule la paie d’un employé salarié, on retourne tout simplement son salaire par semaine.

Quand on calcule la paie d’un employé salarié qui est aussi à commission, on retourne tout simplement son salaire par semaine plus sa commission.

Quand on calcule la paie d’un employé payé à l’heure, on retourne les heures travaillées X le taux horaire.

Finalement, la méthode afficher\_paies() du système de paie prendra une liste d’employés et fera une boucle pour imprimer les infos de chaque employé et le montant net de la paie obtenu par l’appel de la méthode calculer\_paie() de chaque employé.

## Instanciez les objets des différentes classes et faites les tests demandés ci-dessous.

Instanciation des objets

|  |
| --- |
| Action |
| -Créez un nouvel employé salarié, avec l’id 1, le nom ‘Marc Tremblay’ et un salaire de 2100 |
| -Créez un nouvel employé payé à l’heure, avec l’id 2, le nom ‘Pierre Jonhson’, qui travaille 40 heures/semaine au taux horaire de 22$ |
| -Créez un nouvel employé à commission, avec l’id 3, le nom Luc Toupin, qui a un salaire de semaine de 1400 et une commission de 600. |
| -SANS instancier un système de paie, appelez la méthode afficher\_paies() en lui passant en paramètre la liste des employés créé précédemment. |
| Résultat attendu : |

## Remise :

Dans GitHub, faites la remise de votre code python.

Cette remise est à faire AVANT la fin de ce laboratoire.