

TP 3

Héritage et polymorphisme

1 Contenu et objectifs des TPs

Nous allons dans ce TP appliquer l'**héritage** et le **polymorphisme** dans un sujet concret sur les employés d'une entreprise : vous allez programmer le calcul des salaires hebdomadaires des employés d'une entreprise.

NB : sujet adapté de collègues niçois, merci à eux !

2 Mise en oeuvre

2.1 Spécifications de l'entreprise et modélisation :

Cette entreprise comporte plusieurs types d'employés :

1. Des employés qui sont payés suivant le nombre d'heures qu'ils ont travaillé dans la semaine. Ils sont payés à un certain tarif horaire et leurs heures supplémentaires (au-delà de 35 heures) sont payées 30% de plus que les heures normales.
2. D'autres employés, payés de la même façon, mais leurs heures supplémentaires sont payées 50 % de plus que les heures normales.
3. Les commerciaux sont payés avec une somme fixe à laquelle on ajoute 1 % du chiffre d'affaires qu'ils ont fait dans la semaine.

Exercice 1.

Selon les spécifications précédentes et en vous aidant des recommandations ci-dessous, écrivez votre hiérarchie d'héritage **sur papier** et **faites-la vérifier par l'enseignant avant de commencer à coder**.

NB : dans votre hiérarchie, on doit retrouver les classes, leurs attributs, et fonctions ainsi que les liens d'héritage.

- Cette situation peut se modéliser à l'aide de deux classes que nous nommerons **EmployeHoraire** (cas 1 et 2) et **Commercial** (cas 3).
- Vous donnerez un nom à chacun des employés. On ne pourra modifier le nom d'un employé.
- Tout employé ayant un nom, vous utiliserez une (super-)classe **Employe** dont hériteront les autres classes.
- Le temps est compté pour faire ce TP. Pour ne pas avoir trop de modificateurs vous simplifierez les accesseurs en ne donnant qu'un seul modificateur **setInfosSalaire** pour entrer ou modifier les informations brutes nécessaires au calcul des salaires (nombre d'heures de travail, chiffre d'affaires,...). **N'essayez pas de faire du polymorphisme avec cette méthode.**
- Les classes comporteront au moins 2 constructeurs : un qui ne prend en paramètre que le nom de l'employé et l'autre qui prend en paramètres le nom et toutes les informations pour le calcul du salaire.
- Le calcul des salaires se fera dans la méthode **getSalaire()** qui sera utilisée pour **faire du polymorphisme**.

2.2 Création d'une entreprise et utilisation des classes :

Exercice 2.

Pour tester vos classes `EmployeHoraire` et `Commercial`, vous écrirez une classe `Paie` qui comportera une unique méthode `main()` et qui :

- entrera les informations sur des employés des différents types (au moins un de chaque type), avec :
 - au moins un des employés sera créé avec le constructeur qui n'a que le nom en paramètre, et vous entrerez ensuite les informations pour son salaire avec la méthode `setInfosSalaire`.
 - au moins un autre employé sera créé avec le constructeur prenant en paramètre les informations sur le salaire.
- enregistrera les employés dans un tableau d'employés,
- affichera de manière lisible le salaire hebdomadaire de chacun des employés dans une boucle `for` qui parcourra le tableau des employés.
- L'affichage aura la forme : "Dupond gagne 2500 ", avec ajout des informations spécifiques ayant permis de calculer le salaire de chaque type d'employé. Vérifiez les calculs des salaires!