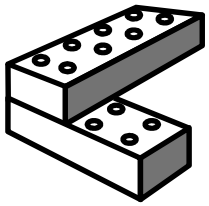


Travaux Pratiques : Enoncés
230-TP02_PHP_POO_SalaireEmploye

Objectif : **Création et utilisation de classes abstraites et dérivées**
Techniques abordées : Création de classe « abstract », de méthode « abstract » et d'héritage

Cdc :

On se propose de calculer le salaire mensuel (très simplifié) de différents statuts d'employés.
Certains employés sont rémunérés à l'heure en ayant une base horaire, un nombre d'heures de base, un nombre d'heures effectivement travaillées et un taux permettant de calculer la rémunération de ses heures supplémentaires.

« salaire = base_horaire *(nbr_heure_de_base + heures_sup * (1 + taux)) »

D'autres employés sont des commerciaux ayant un salaire fixe et une partie basée sur un pourcentage de son chiffre d'affaire. Un taux permet de calculer cette partie non fixe.

« salaire = fixe + chiffre_affaire * taux »

Après avoir rentré les informations servant au calcul du salaire, le système affichera la rémunération mensuelle de l'employé.

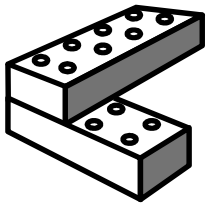
Réalisation :

1 : Créer un « projet PhP» nommé « 230-TP02_PHP_POO_SalaireEmploye ».

2 : Créer un premier fichier nommé « calculSalaire.php » qui servira à tester les différentes classes.

3 : Créer un autre fichier nommé « Employe. php » qui contiendra la définition d'une classe abstraite « Employe » dont la structure est la suivante :

- **Deux propriétés protégées :**
  « nom » (chaine) et « matricule » (entier)
- **Une propriété de classe privée**
  « dernierMatricule » (entier) : elle mémorise le dernier numéro de matricule
- **Un constructeur acceptant en paramètre deux valeurs permettant d'initialiser les propriétés comme suit :**
  nom en majuscule
  matricule par la valeur de « dernierMatricule » incrémenté de 1
- **Une méthode abstraite protégée « getSalaire() » obligeant les classes dérivées à la redéfinir.**
- **Une méthode concrète publique « infosPerso » permettant d'afficher les informations sur l'employé :**
  Employé : "son matricule"
  Nom : "son nom"
- **Une méthode concrète publique « afficheSalaire » qui donne le message suivant :**
  Même affichage que infosPerso + Salaire du mois : "xx.xx"



Travaux Pratiques : Enoncés
230-TP02_PHP_POO_SalaireEmploye

4 : Créer deux autres classes concrètes « EmployeHoraire » (EmployeHoraire.php) et « Commercial » (Commercial.php) héritant de la classe « Employe » avec comme propriétés :

Pour « EmployeHoraire »

- **Une propriété : « nbrHeureDeBase »**
- **Une propritété : « nbrHeureReel »**
- **Une propriété : « salHoraire »**
- **Une propriété static : « taux » initialisée à 10.0 (10 %)**

Pour « Commercial »

- **Une propriété : « ca »**
- **Une propriété : « salFixe »**
- **Une propriété static : « taux » initialisée à 10.0 (10 %)**

Pour les deux classes, redéfinissez :

- **Le constructeur acceptant tous les paramètres**
- **La méthode « getSalaire » retournant le salaire**

5 : Dans « calculSalaire.php »

1. **créer un commercial en fournissant son nom, son salaire fixe, son chiffre d'affaire et son taux puis afficher les résultats.**
2. **Faites de même pour un employé horaire en donnant son nom, sa rémunération horaire, son nombre d'heures de base et son taux puis afficher les résultats.**

6 : Pour la gestion du numéro de matricule, trouvez le moyen d'une gestion automatique qui incrémente de 1 ce numéro à chaque nouvel employé.

<u>Exemple de réalisation :</u>