

Série Exercices POO en PHP

Exercice 1

Créez une classe représentant une personne. Elle doit avoir les propriétés nom, prénom et adresse, ainsi qu'un constructeur et un destructeur. Une méthode `getpersonne()` doit retourner les coordonnées complètes de la personne. Une méthode `setadresse()` doit permettre de modifier l'adresse de la personne. Créez des objets personne, et utilisez l'ensemble des méthodes.

Exercice 2

Créez une classe abstraite représentant une personne. Elle déclare les propriétés nom et prénom et un constructeur. Créez une classe client dérivée de la classe personne en y ajoutant la propriété adresse et une méthode `setcoord()` qui affiche les coordonnées complètes de la personne. Créez une classe électeur dérivée de la même classe abstraite, et ajoutez-y deux propriétés `bureau_de_vote` et `vote` ainsi qu'une méthode `avoter()`, qui enregistre si une personne a voté dans la propriété `vote`.

Exercice 3

Le directeur d'une entreprise de produits chimiques souhaite gérer les salaires et primes de ses employés au moyen d'un programme PHP5.

Un employé est caractérisé par son nom, son ancienneté et le service dans lequel il travaille.

Coder une classe **Employe** dotée des attributs nécessaires, d'un constructeur et d'une méthode **calculerSalaire**. Le salaire dépend de l'ancienneté : il est égal à 400 si l'ancienneté est inférieure à 10 ans, sinon il sera égal à 550.

Cependant, le calcul du salaire dépend aussi du type de service auquel appartient l'employé. On distingue alors les employés affectés au service *Production*. Dans le service Production, une prime sera accordée aux employés. Leur salaire (déjà calculé en fonction de l'ancienneté) sera augmenté par le *nombre d'unités* produites mensuellement multipliées par 5.

Coder une hiérarchie de classes pour les employés en respectant les conditions suivantes :

- La super-classe de la hiérarchie doit être la classe **Employe**.
- La nouvelle classe **EmployeProd** doit contenir les attributs qui lui sont spécifiques ainsi que le codage approprié de la méthode **calculerSalaire**.
- La sous classe est dotée de constructeur prenant en argument l'ensemble des attributs nécessaires.

Terminez l'écriture du script PHP5 pour avoir l'affichage suivant :

