

## Exercices de POO en PHP

### Exercice 5 : L'opérateur de résolutions de portée

Dans cet exercice vous allez créer une classe très simple chargée de représenter un objet que vous connaissez tous ainsi que quelques unes de ses caractéristiques : les Renault Clio.

Vous allez donc déclarer une classe clio dans laquelle nous allons intégrer quelques caractéristiques d'une vraie clio à savoir : le nombre de portes, la couleur de la carrosserie, et le prix neuf de la voiture.

Une fois cette classe avec ses attributs, ses getters et ses setters créée, instanciez plusieurs clio avec des nombres de portes différents et des couleurs différentes et affichez ces valeurs à l'écran.

Attention : l'objectif de cet exercice est de vous faire travailler sur l'opérateur de résolution de portée, c'est à dire les constantes de classe, les attributs statiques et les fonctions statiques. Autrement-dit il faudra bien vous interroger sur ce que vous utilisez, il n'y aura pas que de simples attributs et méthodes dans votre classe.

De plus vous allez vous renseigner sur le mode strict de PHP et l'intégrer à votre script. De cette manière vous pourrez par exemple dire que telle méthode attend explicitement un integer, une string ou un autre types.

Pour vous aider, voici quelques petites informations que vous devriez savoir :

- Une clio n'a que deux nombres de portes possibles : 3 et 5 cela ne change jamais !
- Une clio n'a que 8 couleurs de carrosserie possibles, une couleur est définie par un nom « bleu nuit » par exemple et une référence couleur telle que 213800058. Quand on demande la couleur de la carrosserie avec le getter, c'est cette référence qui est appelée.
- Le prix neuf de la voiture est commun à toute les clios créées, si le prix neuf d'une voiture change alors celui de toutes les voitures change aussi.

### Exercice 6 : gérer ses objets en base de données

Dans cet exercice vous allez gérer une petite base de données en orienté objet, celle-ci vous permettra de gérer des chats.

**Attention, vous n'utiliserez pas le type strict**

étape 1 :

Créez une classe chat avec les attributs suivants :

- Nom
- Age
- Sexe
- Couleur

N'oubliez pas que votre chat ne peut avoir que deux sexes différents, il s'agit donc d'une valeur constante. Idem pour les couleurs. Ces valeurs nous serviront à effectuer des vérifications

N'oubliez également de forcer le typage des arguments dans vos fonctions.

Dans vos setters vérifiez entre autres que :

- Le nom ne dépasse pas 15 caractères
- L'âge est compris entre 0 et 30
- Le sexe appartient aux sexes définis dans la constante de classe
- La couleur appartient aux couleurs définies dans la constante de classe

Etape 2 :

Créez un objet chatManager vous permettant de gérer vos chats en base de données. Cette objet vous permettra notamment :

- d'enregistrer un objet chat
- de sélectionner tous les chats de votre base de données

Etape 3 :

Créez une petite vue contenant un formulaire, quand l'utilisateur remplit ce formulaire, un nouvel objet chat est enregistré en base de données et il apparaît dans la vue qui liste tous les chats enregistrés.

### **Exercice 7 : héritage**

Créez une sous-classe nommée upgradedForm héritant de la classe form de l'exercice 4.

Cette nouvelle classe doit permettre de créer des formulaires ayant en plus des boutons radio et des cases à cocher.

Elle doit donc avoir les méthodes supplémentaires qui correspondent à la création de ces inputs. Créez des objets form et testez le résultat.

### **Exercice 8 : héritage**

Créez une classe abstraite personne. Elle déclare les propriétés :

- nom
- prénom

Créez une classe client héritant de la classe personne en y ajoutant la propriété adresse. Client dispose d'une méthode getCoord() qui affiche les coordonnées complètes de la personne. Elle affiche par exemple « nom + prénom habite au adresse ».

Créez une classe électeur héritant elle aussi de la classe personne, et ajoutez-y deux propriétés bureau (pour bureau de vote), vote, ainsi qu'une méthode aVoter(), qui enregistre si une personne a voté dans la propriété vote et affiche un message de succès.

Si la personne a déjà voté elle affiche un petit message d'erreur.

### **Exercice 9 : héritage**

Etape 1 :

Créez deux classes « voiture » et « bus », ces classes permettent de gérer :

- L'immatriculation du véhicule
- La couleur du véhicule

Cependant chaque classe a ses spécificités, la classe voiture permet en plus de :

- gérer le nombre de portes de la voiture

La classe bus permet de :

- gérer le nombre d'étages du bus

NB : Ces deux classes devront hériter d'une classe commune qui leur permettra de partager certaines caractéristiques.

Le nombre d'étages et de portes sont des valeurs qui ne changent jamais ! Une voiture ne peut pas avoir 1 porte par exemple, de même un bus à 3 étages n'existe pas.

Etape 2 :

Créez une instance de la classe bus et de la classe voiture, donnez leur l'immatriculation et la couleur de votre choix. Indiquez le nombre d'étages de votre choix pour le bus et le nombre de portes de votre choix pour la voiture.

Etape 3 :

Affichez les valeurs des propriétés de chaque véhicule à la suite.

Changez la couleur de la voiture et le nombre d'étages du bus.

Affichez à nouveau les valeurs des attributs