



TP6 – PHP Orienté Objet

Dans ce TP nous travaillerons dans un premier temps sur la création, l'instanciation et la destruction d'objets PHP. Puis, nous traiterons les notions d'héritage entre deux classes d'objets. Nous terminerons par un exemple de création d'objet à partir d'une table contenue dans une base de données et de la mise à jour de cette même table lors de l'instanciation de l'objet.

1 Création, Instanciation et Destruction d'Objets PHP

1.1 Créer une classe « Voiture » qui contient les propriétés suivantes qui sont bien entendues privées :

- peinture (3 valeurs possibles : « classique », « metallisee », « nacree »),
- couleur,
- boite (3 valeurs possibles : « manuelle », « robotisee », « automatique »),
- nb_porte (4 valeurs possibles : 2, 3, 4, 5),
- radar_recul (booléen),
- capteur_pluie (booléen),
- capteur_luminosite (booléen),
- regulateur_vitesse (booléen).

1.2 Ajouter des méthodes publiques à cette classe :

- setPeinture() pour affecter le type peinture avec les contraintes ci-dessus,
- setCouleur() pour affecter la couleur,
- setNbPorte() pour affecter le nombre de portes avec les contraintes ci-dessus,
- setOptions() pour affecter les champs restants,
- displayVoiture() pour afficher les caractéristiques de la voiture.

1.3 Nous voulons maintenant ajouter la propriété `__construct()` en tant que constructeur. Dans cette fonction on suppose par défaut que :

- la peinture est classique,
- la boite est manuelle,
- le nombre de porte est 5,
- les options sont toutes affectées à false.

Bien que ces propriétés sont affectées par défaut le constructeur doit prévoir de les modifier au gré du programmeur. La propriété couleur doit, quand à elle, être désignée par le programmeur quoi qu'il arrive.

1.4 Enfin, nous terminons la définition de la classe par l'ajout de la méthode `_destruct()` pour détruire l'objet qui se contentera d'afficher le message « La voiture de couleur ... et ayant ... portes a été détruite ».

1.5 Créer deux objets « berline » et « coupe » de classe « Voiture » ; la première est de couleur rouge et a 4 portes et la seconde est de couleur bleu et a 2 portes. Vous êtes libre en dehors de cela de choisir les caractéristiques de ces voitures.

- Tester les méthodes définies en 1.2 pour affecter les propriétés des objets voitures.
- Tester la fonction de destruction de l'objet berline.
- Nous faisons l'affectation `coupe2 = coupe`, puis détruisons l'objet coupe, que se passe-t-il, l'objet coupe est-il réellement détruit ? Sinon que faut-il faire ?

2 Héritage

2.1 Nous voulons créer maintenant la classe « Vehicule » qui possède les propriétés suivantes :

- marque
- modele
- prix

Nous souhaitons en outre que ces propriétés soient visibles par les objets qui héritent de l'objet « Vehicule ».

2.2 Ajouter les méthodes permettant de construire l'objet « Vehicule » et d'affecter les propriétés de l'objet à savoir :

- `setMarque()`,
- `setModele()`, sachant que le modèle doit respecter la marque dont il est issu,
- `setPrix()`, sachant que le prix ne peut varier qu'entre 5000 et 300000.

2.3 Redéfinissez la classe « Voiture » afin qu'elle hérite de la classe « Vehicule ».

2.4 Modifiez le constructeur de la classe voiture afin qu'il tienne compte de la marque, du modèle et du prix du véhicule. Modifier la fonction d'affichage des caractéristiques d'une voiture pour qu'elle tienne compte de sa marque, de son modèle et de son prix.

2.5 Comment peut-on modifier les champs d'un objet « Vehicule » à partir d'un objet « Voiture » et des méthodes de ce même objet « Vehicule » ?

2.6 Créez un objet « Vehicule » de marque « BMW », de modèle « Z4 » et de prix 35000. Les autres propriétés de cet objet seront définies à votre guise. Affichez l'objet ainsi construit.

3 Objets et Tables

Cette partie du TP est relative à la génération d'objets à partir de table et réciproquement à l'insertion ou à la mise à jour des données d'une table lorsqu'un objet correspondant est créé ou modifié.

3.1 Créer une table « Vehicule » constituée par les propriétés de la classe « Vehicule ». Cette table contient les données relatives aux marques « Renault », « Peugeot » et « Volkswagen » et respectivement aux modèles (« Twingo », « Clio », « Megane »), (« 107 », « 207 », « 308 ») et (« Lupo », « Polo », « Golf »). Puis créer une table « Voiture » constituée par les propriétés de la classe « Voiture », en outre cette table contient :

- un champ d'identifiant unique « id_voiture », qui est incrémenté automatiquement de 1 lors de l'insertion d'une nouvelle voiture,
- un champ « marque »,
- un champ « modele »,
- un champ « prix ».

3-2 Modifier le constructeur de la classe « Vehicule » afin de créer l'ensemble des objets de classe « Vehicule » d'une marque donnée à partir de la table « Vehicule ».

3-3 Modifier les propriétés et le constructeur de la classe « Voiture » afin de modifier la table « Voiture » lors de la création d'un nouvel objet « Voiture ».

3-4 Modifier les méthodes de la classe « Voiture » afin de mettre à jour la table « Voiture » lors de la modification d'un objet « Voiture ».