

Fiche de révision

Les Classes en PHP

1. Qu'est-ce que c'est ?

Une classe en programmation orientée objet (POO) est un plan ou un modèle qui définit les caractéristiques et les comportements (attributs et méthodes) que les objets créés à partir de la classe auront. La classe est l'une des fondations de la POO, car elle permet de regrouper des données et des fonctionnalités de manière structurée.

Voici les éléments clés d'une classe en POO :

- **Attributs** (Propriétés) : Ce sont les variables définies dans une classe qui détiennent les états ou les données. Les attributs représentent les caractéristiques des objets que la classe va créer. Par exemple, pour une classe « Voiture », les attributs pourraient inclure «marque», «modèle», et «couleur».
- **Méthodes** (Fonctions) : Les méthodes définies dans une classe décrivent les comportements ou les actions que les objets de la classe peuvent effectuer. Elles agissent sur les données (attributs) et peuvent exécuter toute logique nécessaire. Reprenant l'exemple de la classe « Voiture », les méthodes pourraient inclure « demarrer() », « arreter() », et «changerVitesse()».
- **Instances** : Une classe sert de modèle pour créer des instances (objets), chaque objet ayant son propre état indépendant mais partageant la structure et les comportements définis par la classe.

En résumé, une classe en programmation orientée objet est un concept fondamental qui encapsule des données et des fonctionnalités, fournissant un moyen de modéliser des entités du monde réel ou des concepts abstraits de manière structurée et réutilisable.

“Deleted code is debugged code” - Jeff Sickel

2. Comment faire ?

Créer une classe en PHP est un processus fondamental dans la programmation orientée objet (POO) qui vous permet de structurer votre code de manière modulaire et réutilisable. Voici un guide étape par étape pour créer une classe en PHP.

2.1. Définir la classe

Utilisez le mot-clé `class` suivi du nom de la classe. Le nom de la classe doit commencer par une lettre majuscule par convention.

```
class Voiture {  
    // Le corps de la classe où vous définirez les propriétés et les méthodes  
}
```

2.2. Définir les propriétés

Définissez les attributs ou propriétés de la classe. Ces propriétés stockeront les données ou l'état de vos objets. Le modificateur d'accès `public`, permet de donner une visibilité aux propriétés sans limites.

```
class Voiture {  
    public string $marque;  
    public string $modele;  
    public int $vitesse;  
}
```

2.3. Créer un constructeur

Le constructeur est une méthode spéciale appelée lorsque vous créez une instance de la classe. Il est souvent utilisé pour initialiser les propriétés de l'objet. Cette méthode est appelée quand on écrit le mot-clé « `new` ».

```
class Voiture {  
    public string $marque;  
    public string $modele;  
    public int $vitesse;  
  
    public function __construct(string $marque, string $modele, int $vitesse) {  
        $this->marque = $marque;  
        $this->modele = $modele;  
        $this->vitesse = $vitesse;  
    }  
}
```

2.4. Ajouter des méthodes

Définissez les comportements de votre classe à l'aide de méthodes. Les méthodes agissent sur les propriétés de l'objet et définissent les actions que les objets de la classe peuvent effectuer.

```
class Voiture {  
    public string $marque;  
    public string $modele;  
    public int $vitesse;  
  
    public function __construct(string $marque,string $modele,int $vitesse) {  
        $this->marque = $marque;  
        $this->modele = $modele;  
        $this->vitesse = $vitesse;  
    }  
    public function afficherDetails() :string {  
        echo "Marque: " . $this->marque . ", Modèle: " . $this->modele;  
    }  
}
```

2.5. Instancier la classe

Créez un objet de la classe en utilisant le mot-clé **new** suivi du nom de la classe et passez les arguments nécessaires au constructeur.

```
$maVoiture = new Voiture("Toyota", "Corolla",160);  
$maVoiture->afficherDetails(); // Affiche "Marque: Toyota, Modèle: Corolla"
```

3. Bonnes Pratiques

Voici quelques bonnes pratiques à se rappeler :

- **Nommez clairement les classes et les méthodes** : Utilisez des noms descriptifs pour les classes, les méthodes et les propriétés pour rendre votre code plus lisible.
- **Principe de responsabilité unique** : Chaque classe doit avoir une seule responsabilité. Ne surchargez pas une classe avec trop de fonctionnalités.
- **Typage** : typer les attributs, les paramètres et donnez un type de retour pour toutes les méthodes (void pour les méthodes ne retournant rien = le vide).