

# PROGRAMMATION ORIENTÉE OBJET EN PHP

Document / 12/15/2023

Préparé par

NABIL CHABAB

Approuvé par

MR SAID/TO BE OR NOT TO BE

A decorative graphic consisting of multiple thin, light blue wavy lines that flow from the bottom right towards the center of the page, creating a sense of movement and depth.

## INTRODUCTION

La programmation orientée objet (POO) est un paradigme de programmation qui utilise des objets pour structurer le code. En PHP, la POO offre des fonctionnalités puissantes pour organiser et réutiliser du code de manière efficace.

## LES BASES

### Classes:

**Une classe est un modèle pour créer des objets. Elle définit les propriétés et les méthodes que les objets auront.**

**Exemple de classe en PHP :**

```
class Personne {  
    // Propriétés  
    public $nom;  
    public $age;  
  
    // Méthodes  
    public function afficherDetails() {  
        echo "Nom: $this->nom, Age: $this->age";  
    }  
}
```

## CRÉATION D'OBJETS

Les objets sont des instances de classes. Pour créer un objet en PHP, utilisez le mot-clé `new`.

```
// Création d'un objet Personne
$personne1 = new Personne();
$personne1->nom = "John Doe";
$personne1->age = 30;

// Appel de la méthode pour afficher les détails
$personne1->afficherDetails();
```

## ENCAPSULATION

L'encapsulation consiste à regrouper les propriétés et les méthodes pertinentes dans une classe, puis à restreindre l'accès à certaines propriétés ou méthodes.

```
class CompteBancaire {
    // Propriétés encapsulées
    private $solde;

    // Méthodes publiques pour accéder au solde
    public function getSolde() {
        return $this->solde;
    }

    public function deposer($montant) {
        $this->solde += $montant;
    }
}
```

## HÉRITAGE

L'héritage permet à une classe (appelée classe dérivée) d'hériter des propriétés et des méthodes d'une autre classe (appelée classe de base).

```
class Employe extends Personne {  
    // Nouvelle propriété spécifique à la classe Employe  
    public $salaire;  
  
    // Méthode spécifique à la classe Employe  
    public function afficherSalaire() {  
        echo "Salaire: $this->salaire";  
    }  
}
```

## ABSTRACTION

L'abstraction consiste à définir une classe de manière à ce qu'elle puisse être utilisée comme un modèle générique, en cachant les détails d'implémentation.

```
abstract class Forme {  
    // Méthode abstraite  
    abstract public function calculerSurface();  
}
```

```
class Cercle extends Forme {  
    private $rayon;  
  
    public function __construct($rayon) {  
        $this->rayon = $rayon;  
    }  
  
    public function calculerSurface() {  
        return pi() * pow($this->rayon, 2);  
    }  
}
```

## INTERFACES

**Les interfaces définissent un ensemble de méthodes que les classes doivent implémenter. Elles permettent de définir un contrat pour les classes.**

```
interface Animal {  
    public function faireDuBruit();  
}  
  
class Chien implements Animal {  
    public function faireDuBruit() {  
        echo "Woof! Woof!";  
    }  
}
```

**Ces concepts de POO en PHP vous permettront de créer un code bien structuré, réutilisable et facile à maintenir. N'hésitez pas à explorer davantage chaque concept pour approfondir votre compréhension**

## CONCLUSION

**En conclusion, la programmation orientée objet (POO) en PHP offre un moyen puissant d'organiser et de structurer le code de manière à le rendre plus modulaire, réutilisable et facile à comprendre. Nous avons exploré les concepts fondamentaux de la POO, tels que les classes, la création d'objets, l'encapsulation, l'héritage, l'abstraction et les interfaces.**

**Classes et Objets : Les classes servent de modèles pour créer des objets, regroupant propriétés et méthodes liées. Les objets sont des instances concrètes de ces classes.**

**Encapsulation : En encapsulant les propriétés et les méthodes, nous limitons l'accès à certaines parties du code, renforçant ainsi la sécurité et la maintenance.**

**Héritage : L'héritage permet de créer des classes dérivées qui héritent des propriétés et des méthodes d'une classe de base, favorisant la réutilisation du code.**

**Abstraction : L'abstraction permet de créer des modèles génériques en cachant les détails d'implémentation, facilitant la compréhension et la maintenance du code.**

**Interfaces : Les interfaces définissent des contrats que les classes doivent suivre, favorisant la normalisation et l'interopérabilité entre différentes parties du code.**

**La POO encourage la modularité, la réutilisation et la séparation des préoccupations, améliorant ainsi la maintenabilité des applications. En continuant d'explorer ces concepts et en les appliquant dans vos projets, vous serez en mesure de créer un code plus robuste et extensible en PHP. Bonne programmation orientée objet !**

**YouCode**

