

## Fiche de la formation

**Durée :** 2 jours

### Audience :

Développeurs PHP confirmés.

### Pré-requis :

Bonne connaissance du langage PHP.

### Objectif pédagogiques :

Développer la capacité à exploiter des objets existants, à concevoir des systèmes d'objets et à exploiter le meilleur de PHP.

### Méthodologie :

La méthodologie mise en oeuvre dans cette formation a pour but premier une prise en main réelle des outils et technologies abordés, orientée vers une maîtrise propre à assurer une productivité en accord avec les besoins de l'entreprise.

## En bref

Le savoir est la composante essentielle du savoir-faire !

Pour que vos équipes soient performantes sur une technologie ou un outil, elles doivent en connaître les fondamentaux. L'expérience démontre que les meilleurs outils utilisés sans formation initiale sont très largement sous-exploités, voire contre-productif.

Afin d'éviter cette situation, génératrice de frustration et obstacle à la qualité, Vesperia vous propose des prestations de transfert de compétence à forte valeur ajoutée, reposant soit sur le matériel de formation des technologies concernées, soit sur des programmes établis sur-mesure (suite à un audit de compétences).

## Description

Les méthodes de développement modernes permettent plus de contrôle dans l'architecture des applications et dans la qualité du code produit.

Pour exploiter ces techniques, il est nécessaire de maîtriser le paradigme objet, selon lequel toutes les applications sont désormais conçues.

Maîtriser la programmation orientée objet apporte au développeur un gain de productivité et de confort significatif, au service des applications qu'il est en charge de réaliser.

## Programme de cours de la formation

### Programmation Orientée Objet

1. Introduction au cours
2. Qu'est-ce que la POO ?
3. Les bases de la POO
4. Mécanismes d'héritage
5. POO avancée

## Plan de formation détaillé

### 1. Introduction au cours

1. Règles
2. Méthodologie
3. Objectifs

### 2. Qu'est-ce que la POO ?

1. Paradigmes
  1. Procédural
  2. Orientée Objet
2. Avantages de la POO
  1. Donner du sens
  2. Etendre
  3. Factoriser
  4. Tester unitairement

### 3. Les bases de la POO

1. Notions fondamentales
  1. Classe et instance
  2. Propriété
  3. Méthode
  4. Constante
  5. Visibilité
2. Utilisation d'objets
  1. Instanciation
  2. Accès aux membres
  3. Identification
  4. Typage de paramètres
3. Conception d'un objet
  1. Déclaration de la classe
  2. Définir ses membres
  3. Conception d'un système d'objet
    1. Espaces de noms
    2. Hiérarchisation

### 4. Mécanismes d'héritage

1. Héritage simple
  1. Etendre une classe
  2. Identifier l'ascendance d'une classe
  3. Déterminer la visibilité des membres
2. Abstraction
  1. Déclarer une classe abstraite
  2. Définir des méthodes abstraites
3. Contractualisation
  1. Définir une interface
  2. Différences avec les classes abstraites

### 5. POO avancée

1. La gestion d'erreur par exception
2. Les méthodes magiques
3. La classe Closure
4. La SPL
5. Héritage horizontal
  1. Les traits
  2. Déclaration d'un trait
  3. Utilisation d'un trait dans une classe
  4. Gestion des collisions de nommage