



Développement web coté serveur

AU : 2024-2025

Enseignante : CHERIF Nidhal



Atelier N°5

POO(Programmation Orientée Objet)

Objectifs :

- Appliquer les notions de l'orienté objet avec PHP : Classe, héritage, encapsulation, Getter et Setter,
- Définir et utiliser méthodes magiques de php (`__set()`, `__get()`, `__destruct()`, `__construct()`, `__toString()`)

EXERCICE 1

- 1) Définir une classe représentant une ville Suivante :

Ville	
-Nom	(string)
-Gouvernorat	(string)
+construct(\$n,\$g)	
+get_info()	(string)

- La méthode **get_info()** permet de retourner un texte comme suit : « La ville X est dans le gouvernorat Y »
- 2) Créer deux objets \$ville1 et \$ville2 ayant respectivement les noms « Chebba » et « Rades » et les gouvernorats « Mahdia » et « Ben Arous »
 - 3) Utiliser la méthode `get_info()` pour les deux objets ;
 - 4) Remplacer la méthode **get_info()** par la méthode magique **__toString()** et faire le test avec `echo $ville1` ;

EXERCICE 2

Créer la classe MiseEnForme qui contient :

- Une propriété privée nommée chaîne ;
- quatre méthodes de type public nommées `gras()`, `italique()`, `souligne()` et `majuscules()` qui retournent respectivement la chaîne passée en argument mise en gras, en italique, soulignée et mise en majuscules.

Définissez un script PHP qui crée un objet basé sur la classe MiseEnForme qui l'exploite pour obtenir le résultat suivant :

Gras : Programmation orientée objet en PHP

Italique : Programmation orientée objet en PHP

Souligné : Programmation orientée objet en PHP

Majuscules : PROGRAMMATION ORIENTÉE OBJET EN PHP

EXERCICE 3 :

Un salarié est décrit par ses propriétés : un code, un nom, un prénom, un salaire de base et une indemnité.

On dérive la classe **SALARIE** par une classe **CONTRACTUEL** : en plus des propriétés du salarié, la classe Contractuel a comme propriétés : la date du contrat et le type du contrat

2.1. Définir la classe **SALARIE** avec

- un constructeur qui initialise les différentes propriétés
- une méthode **calculSalaire** : qui retourne le salaire calculé d'un salarié sachant que : $\text{salaire} = \text{salaire de base} + \text{indemnités}$.
- Une méthode **infosSalarie** : qui retourne une chaîne de caractères contenant les informations d'un salarié (code, nom, prénom, salaire de base et indemnité).

2.2. Construire la classe fille : **CONTRACTUEL**

- Redéfinir le constructeur dans la classe **CONTRACTUEL**, permettant d'initialiser les différentes propriétés, en utilisant le constructeur parent
- Redéfinir la méthode **infosSalarie** dans la classe **CONTRACTUEL** afin de retourner tout le détail d'un contractuel sous forme d'une chaîne, en faisant appel à la méthode **infosSalarie** de la classe mère **SALARIE**

EXERCICE 4 :

1. Créez une classe abstraite **Personne** avec les propriétés protégées : Nom et Prénom, un constructeur à deux paramètres et une méthode abstraite **getInfo()** qui return les informations des objets des classes filles

2. Créez deux classes filles:

- Adulte: métier, salaire et un constructeur
- Adolescent: âge, loisir et un constructeur
- Créer deux objets \$Adulte1, \$Adolescent1 et tester la méthode getInfo()