

Module Programmation Web/serveur S3

TD n°2 : classes et objets

Tout le code produit doit être conforme au standard PSR-1.

Les types des arguments et les types de retour doivent être précisés (optionnel).

Exercice 1

Construire une classe pour représenter des Personnes : nom, prénom, âge, adresse, ville, code postal, mail, tel. mobile, identifiant skype.

- Tous les attributs sont publics,
- Prévoir un constructeur recevant le nom (uniquement) en paramètre,
- Programmer une méthode nommée `__toString()` qui **retourne** une chaîne de caractère décrivant la personne courante, au format JSON : `{ "<nomatt>" : <valatt>, ... }`
- Nommer le fichier `Personne.php`

Exercice 2

Construire un programme utilisant cette classe `Personne`. Faites un fichier séparé et faites un `require` pour utiliser la classe.

Le programme :

- Créé 2 personnes, et valorise les attributs,
- Affiche des informations sur chacune des 2 personnes (afficher les valeurs des différents attributs). Pour l'affichage, utiliser les constructions et fonctions `echo`, `print`, `printf`
- Affiche les objets complets (sans accéder aux attributs) : (a) en utilisant la méthode `__toString()`, (b) sans utiliser cette méthode, avec `echo` puis `print_r` puis `var_dump`

Exercice 3

Construire une classe nommée `AfficheurDePersonne` dont le rôle est de gérer les différentes façons d'afficher une personne.

- Le constructeur reçoit un objet de type `Personne`
- La méthode `vueCourte()` qui **retourne** une chaîne HTML correspondant à un affichage court (nom, prénom, ville) de la personne. La chaîne HTML n'est pas une page complète, mais juste 1 bloc d'affichage (div).
- La méthode `vueDetail()` qui **retourne** une chaîne HTML correspondant à un affichage détaillé (nom, prénom, age, ville, adresse, mail, tel, idSkype) de la personne. La chaîne HTML n'est pas une page complète, mais juste 1 bloc d'affichage (div).
- La méthode `afficher($sel)` qui reçoit en paramètre un sélecteur permettant de choisir entre un affichage court ou détaillé, et réalise l'affichage complet correspondant.
- Utiliser votre afficheur de personnes dans votre programme de la Q2.

Exercice 4 : un soupçon d'intelligence pour la personne

1. Programmer une méthode qui fait compter la personne – de 0 à son âge, puis de son âge à 0. La méthode **affiche** les valeurs.

2. Écrire une méthode pour aider la personne à écrire ses punitions scolaires : elle reçoit en paramètres 1 phrase et un nombre de répétitions et **affiche** cette phrase le nombre désiré de fois.
3. Écrire une méthode qui reçoit en paramètre un nombre entier N et **retourne** l'âge de la personne à l'année N. Pour cela, aidez-vous des fonctions `date()` et `intval()` que vous trouverez dans la documentation officielle.
4. Modifier la classe `Personne` pour représenter son éventuel/éventuelle conjoint(e). Bien entendu il s'agit d'une autre personne, qui peut ne pas exister. Ajouter une méthode pour définir le conjoint. Adapter l'afficheur : prévoir d'afficher une vue courte du conjoint dans la vue détaillée d'une personne.

Exercice 5 : Documentation

Documenter votre code et générez la documentation à l'aide de l'outil apigen.
Pour cela, voir les normes de documentation dans la page Arche.

Note d'octobre 2017 : A ce jour, il semble que apigen ne supporte pas le typage des retours dans les signature des méthodes, avec la notation introduit en PHP7. Il tient compte de l'annotation `@return` dans le commentaire.