TP N°4 : POO en PHP

Objectif

Le but de ce TP est de mettre en œuvre les concepts de la POO avec PHP.

Préparation

- Pour chacune des classes qui seront demandées par la suite, vous créerez un fichier *php* ayant pour nom le même que la classe.
- Toutes les classes (fichiers) doivent être stockées dans un sous dossier ***myClasses***.
- Pour l'auto-chargement des classes, créer un fichier autoload.php dont le code permettra de chargement automatiquement les classes au besoin.
- Créer un fichier ***useClasses.php*** qui doit contenir les scripts d'utilisation des classes définies.
- Séparer les exercices en différentes sections HTML.

Travail à faire

Exercice 1

Définir une classe Eq2Degre avec les caractéristiques suivantes :

- la classe possède les attributs suivants : ***trois réels (a, b et c)*** correspondant aux coefficients du polynôme et un tableau ***(solutions)*** pour contenir les éventuelles solutions dans R.
- définir un constructeur à trois paramètres réels qui correspondent aux coefficients du polynôme à résoudre. Ce constructeur affectera les valeurs passées en paramètre aux attributs a, b et c;
- définir une méthode ***getDiscrimant*** qui calcule et renvoie la valeur du discriminant,
- définir une méthode ***afficheDiscriminant*** qui affiche la valeur du discriminant,
- définir une méthode statique ***racine*** qui renvoie la racine carré du réel passé en paramètre,
- définir une méthode ***resoudre*** qui résout l'équation dans R et initialise l'attribut *solutions* (tableau qui peut être vide);
- définir une méthode ***afficheSolutions*** qui appelle la méthode ***resoudre*** pour afficher les résultats de la résolution de l'équation (*solutions*).
- Instancier votre classe avec des coefficients saisis par l'utilisateur via un formulaire
- Mettre en œuvre toutes ses méthodes.

Exercice 2

Écrire une classe pour définir une **Ville**.

- Elle doit avoir les propriétés nom, département, région et nombre d'habitants.
- En plus des accesseurs (getters), des mutateurs (setters) et du constructeur, définir une méthode qui affiche toutes les informations d'une ville.
- Créez des objets ville, affectez leurs propriétés, et utilisez la méthode d'affichage

Créer un **formulaire de saisie** d'une ville.

- Pour le traitement du formulaire, utiliser les informations récupérées pour instancier une ville et l'afficher par la suite.

Exercice 3

1. Créez une classe représentant une personne. Elle déclare les propriétés nom et prénom et un constructeur. Elle dispose également de méthode pour afficher toutes les informations sur une personne.
2. Créez une classe client dérivée de la classe personne en y ajoutant les propriétés adresse et téléphone une méthode **getCoord()** (utilisant des méthodes de la classe parente) qui

affiche les coordonnées complètes de la personne. Cette classe devra disposer de son constructeur.

3. Créez une classe Electeur dérivée de la même classe Personne, et y ajoutez deux propriétés bureau_de_vote et vote (on considère que par défaut q'un électeur n'a pas voté), ainsi qu'une méthode avoter() , qui enregistre si une personne a voté dans la propriété vote . Cette classe devra disposer de son constructeur.
4. Créer des objets des classes Personne, Client et Electeur pour mettre en œuvre les méthodes et les fonctionnalités sur ces classes

Exercice 4

Écrire une classe pour définir une **Bien**.

- Elle doit avoir les propriétés nécessaires pour caractériser un bien immobilier.
- En plus des accesseurs (getters), des mutateurs (setters) et du constructeur, définir une méthode qui affiche toutes les informations d'un bien immobilier.
- utiliser les informations récupérées du formulaire de saisie d'un bien pour instancier la classe Bien et mettre en œuvre ses méthodes.

Exercice 5

1. Créez les classes **Appartement, Immeuble et terrain** représentant respectivement un appartement, un immeuble et un terrain d'une agence immobilière qui héritent toutes de la classe **Bien** défini dans l'exercice 2°). Compléter les attributs nécessaires, constructeur et méthodes pour chaque classe.
2. Instancier et mettre en œuvre toutes ces classes en utilisant des formulaires.