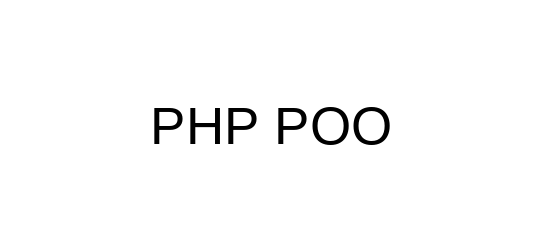
****



Consignes :

Vous devrez me rendre le code commenté ainsi que le lien Github et un petit rapport de tout ce que vous avez utilisé pour le projet dans votre groupe le tout dans un fichier .zip portant votre « nom+prénom+classe-php-Examen.zip ».

Pour ce projet, vous pouvez être en groupe de 2 maximum.

Vous devrez envoyer ce fichier à l’adresse mail suivante :

[j.dubois@ecole-ipssi.net](mailto:j.dubois@ecole-ipssi.net)

Sur ce, bon courage !

Héritage en PHP:

**Objectif :** Mettre en pratique les concepts d'héritage en PHP 8.

**Exercice 1: Création d'une Hiérarchie d'Animaux**

1. **Définition des Classes :**
   * Créez une classe de base **Animal** avec les propriétés communes à tous les animaux (par exemple, nom, âge).
   * Ajoutez une méthode **crier()** qui affiche un message générique.
2. **Classes Filles :**
   * Créez au moins deux classes filles, par exemple **Chien** et **Chat**, qui héritent de la classe **Animal**.
   * Ajoutez des propriétés spécifiques à chaque classe fille.
   * Surchargez la méthode **crier()** dans chaque classe fille pour afficher un cri spécifique à l'animal.
3. **Utilisation des Classes :**
   * Instanciez des objets de chaque classe et appelez la méthode **crier()** pour voir le comportement polymorphique.

**Exercice 2: Modélisation d'un Parc Animalier**

1. **Nouvelles Classes :**
   * Créez une nouvelle classe **ParcAnimalier** qui aura un tableau pour stocker différents animaux.
2. **Ajout d'Animaux au Parc :**
   * Ajoutez quelques animaux de chaque type (Chien, Chat) au parc.
   * Utilisez la méthode **crier()** pour afficher les cris spécifiques.
3. **Affichage de l'Information :**
   * Ajoutez une méthode **afficherInfos()** dans la classe **ParcAnimalier** qui parcourt le tableau d'animaux et affiche les informations de chaque animal.

**Exercice 3: Héritage dans un Contexte Réel**

1. **Définition d'une Classe Parente :**
   * Créez une classe parente **Véhicule** avec des propriétés communes (par exemple, marque, modèle).
   * Ajoutez une méthode **demarrer()**.
2. **Classes Filles :**
   * Créez deux classes filles, par exemple **Voiture** et **Moto**, qui héritent de la classe **Véhicule**.
   * Ajoutez des propriétés spécifiques à chaque classe fille.
   * Surchargez la méthode **demarrer()** pour afficher un message spécifique à chaque type de véhicule.
3. **Utilisation des Classes :**

* Instanciez des objets de chaque classe et appelez la méthode **demarrer()** pour voir le polymorphisme en action.

**Bonus (Facultatif):**

* Ajoutez des propriétés statiques et constantes dans certaines classes.
* Expérimentez avec l'héritage multiple en utilisant des interfaces.
* Créez des méthodes abstraites dans la classe parente et implémentez-les dans les classes filles.

**Consignes :**

* Organisez votre code de manière claire en utilisant des commentaires.
* Assurez-vous que votre code respecte les bonnes pratiques d'utilisation de l'héritage.
* Testez différentes situations pour vous assurer que l'héritage fonctionne correctement.

**À Rendre :**

* Votre ensemble de classes PHP.
* Un petit rapport décrivant les choix de conception que vous avez faits et expliquant comment vous avez testé le polymorphisme.
* Le git sur lequel vous avez mit votre travail.

Haut du formulaire