

Exercice 1

Créez un programme en PHP pour modéliser des formes géométriques.

1. Créez une classe parent **Forme** avec :
 - Un attribut **\$couleur** (protected).
 - Un constructeur pour initialiser la couleur.
 - Une méthode **getCouleur()** qui retourne la couleur de la forme.
2. Créez une classe enfant **Cercle** qui hérite de **Forme** et ajoute :
 - Un attribut **\$rayon** (privé).
 - Une méthode **calculerSurface()** qui retourne la surface du cercle ($\pi \times \text{rayon}^2$).
3. Créez une classe enfant **Rectangle** qui hérite de **Forme** et ajoute :
 - Deux attributs **\$longueur** et **\$largeur** (privés).
 - Une méthode **calculerSurface()** qui retourne la surface du rectangle (longueur \times largeur).
4. Créez un cercle et un rectangle, puis affichez leur couleur et leur surface.

Exercice 2

Créez un programme en PHP pour modéliser un système de gestion de véhicules. Voici les consignes :

1. Créez une classe parent appelée **Vehicule** ayant les attributs suivants :
 - **\$marque** : chaîne de caractères, protected.
 - **\$modele** : chaîne de caractères, protected.
 - **\$annee** : entier, protected.
 - **\$kilometrage** : entier, protected.
2. Et les méthodes :
 - Un constructeur pour initialiser les attributs.
 - Des getters et setters pour chaque attribut.
 - Une méthode **afficherInformations()** qui retourne les informations du véhicule sous forme de chaîne.
3. Créez une classe enfant appelée **Voiture** qui hérite de **Vehicule**. Cette classe doit ajouter :
 - Un attribut **\$nombrePortes** (entier).
 - Une méthode **afficherInformations()** qui surcharge la méthode parent pour inclure le nombre de portes.
4. Créez une classe enfant appelée **Moto** qui hérite de **Vehicule**. Cette classe doit ajouter :
 - Un attribut **\$typeGuidon** (chaîne de caractères).
 - Une méthode **afficherInformations()** qui surcharge la méthode parent pour inclure le type de guidon.

5. Créez plusieurs objets `Voiture` et `Moto`, puis affichez leurs informations en utilisant la méthode `afficherInformations()`.
- 5.

Exercice 3

Créez un programme pour gérer des employés et des managers.

1. Créez une classe parent `Employe` avec :
 - Un attribut statique `$nbEmployes` initialisé à `0` pour compter le nombre total d'employés.
 - Un attribut `$nom` (protected) pour stocker le nom de l'employé.
 - Un constructeur pour initialiser le nom et incrémenter `$nbEmployes`.
 - Une méthode statique `getNbEmployes()` pour retourner le nombre total d'employés.
2. Créez une classe enfant `Manager` qui hérite de `Employe` :
 - Ajoutez un attribut `$equipe` (privé) pour stocker les noms des employés sous la supervision du manager.
 - Ajoutez une méthode `ajouterEmploye($employe)` pour ajouter un employé à l'équipe.
3. Créez plusieurs employés et un manager, affichez le nombre total d'employés et la liste des employés d'un manager.