

Exercices centrés sur les traits en PHP :

Exercice 1 : Création d'un Trait

Énoncé :

Créez un trait `Logger` qui contient deux méthodes : `logInfo($message)` et `logError($message)`. Utilisez ce trait dans deux classes `FichierLogger` et `ConsoleLogger`.

Instructions :

- Définissez le trait `Logger` avec les méthodes `logInfo($message)` et `logError($message)`.
 - `logInfo()` doit afficher le message avec le préfixe "INFO: ".
 - `logError()` doit afficher le message avec le préfixe "ERROR: ".
 - Implémentez la classe `FichierLogger` qui utilise le trait `Logger` et ajoute une méthode `enregistrerDansFichier($message)` pour simuler l'enregistrement d'un message dans un fichier.
 - Implémentez la classe `ConsoleLogger` qui utilise également le trait `Logger` et ajoute une méthode `enregistrerDansConsole($message)` pour simuler l'affichage d'un message dans la console.
 - Créez des instances des deux classes et appelez les méthodes de logging.
-

Exercice 2 : Utilisation de Plusieurs Traits

Énoncé :

Créez deux traits `EmailSender` et `SmsSender`, chacun contenant une méthode pour envoyer un message. Créez une classe `Notification` qui utilise les deux traits.

Instructions :

- Définissez le trait `EmailSender` avec la méthode `envoyerEmail($destinataire, $message)` qui affiche "Email envoyé à \$destinataire: \$message".
 - Définissez le trait `SmsSender` avec la méthode `envoyerSms($numero, $message)` qui affiche "SMS envoyé à \$numero: \$message".
 - Implémentez la classe `Notification` qui utilise les traits `EmailSender` et `SmsSender`.
 - Créez une instance de `Notification` et appelez `envoyerEmail()` et `envoyerSms()` avec des valeurs d'exemple.
-

Exercice 3 : Trait avec Conflit de Méthodes

Énoncé :

Créez deux traits `TraitA` et `TraitB`, chacun contenant une méthode `afficher()`. Dans une classe `MonObjet`, utilisez les deux traits et résolvez le conflit de méthodes.

Instructions :

1. Définissez le trait `TraitA` avec la méthode `afficher()` qui retourne "Affichage depuis TraitA".
2. Définissez le trait `TraitB` avec la méthode `afficher()` qui retourne "Affichage depuis TraitB".
3. Implémentez la classe `MonObjet` qui utilise les deux traits.
4. Résolvez le conflit en redéfinissant la méthode `afficher()` dans `MonObjet` pour qu'elle affiche "Affichage personnalisé".
5. Créez une instance de `MonObjet` et appelez la méthode `afficher()`.