

Fiche de révision

Le Polymorphisme d'Héritage

1. Qu'est-ce que c'est ?

Le polymorphisme est un concept fondamental en programmation orientée objet (POO) qui permet aux objets de différentes classes d'être traités comme des instances d'une même classe parent.

Il offre une manière flexible de travailler avec différents types d'objets qui partagent une classe parente, tout en ayant leur propre implémentation spécifique de certaines méthodes.

Polymorphisme d'Héritage : Cela se produit lorsque différentes classes enfants héritent d'une même classe parente et surchargent ses méthodes.

“Substitutability is a principle in object-oriented programming stating that, in a computer program, if S is a subtype of T, then objects of type T may be replaced with objects of type S without altering any of the desirable properties of the program.”

“Programming made the impossible possible. You have the null object and the constant variable.”

2. Comment faire ?

Il faut mettre en place l'héritage entre la classe « Animal » et deux classes enfants « Chat » et « Chien »

2.1. Création de la classe parent

```
class Animal {  
    public function crier() {  
        echo "L'animal fait un bruit.";  
    }  
}
```

2.2. Création d'une première classe enfant

La classe « Chat » hérite de la classe « Animal » et surcharge la méthode « crier () ».

```
class Chat extends Animal {  
    public function crier() {  
        echo "Le chat miaule.";  
    }  
}
```

2.3. Création d'une deuxième classe enfant

La classe « Chien » hérite de la classe « Animal » et surcharge la méthode « crier () »

```
class Chien extends Animal {  
    public function crier() {  
        echo "Le chien aboie.";  
    }  
}
```

2.4. Création d'une fonction utilisant la classe parent

La fonction « faireCrier () » demande en paramètre un objet de la classe parent « Animal ».

```
function faireCrier(Animal $animal) {  
    $animal->crier();  
}
```

2.5. Utilisation de la fonction avec des objets des classes enfants

Dorénavant, vous pouvez instancier des objets d'une classe enfant et les donner en paramètres dans toute méthode ou fonction demandant un objet de la classe parent « Animal » : chien est une sorte de Animal, chat est une sorte de Animal.

```
$chien = new Chien();  
$chat = new Chat();  
  
faireCrier($chien); // Affiche "Le chien aboie."  
faireCrier($chat); // Affiche "Le chat miaule."
```

3. Bonnes Pratiques

- Utilisez le polymorphisme pour améliorer la flexibilité de votre code en permettant aux fonctions et méthodes de travailler avec des objets de types différents, tant qu'ils partagent une interface ou une classe parente commune.
- Respectez le principe de substitution de Liskov, qui stipule que les objets d'une classe parente doivent pouvoir être remplacés par des objets de classes enfants sans affecter le fonctionnement correct du programme.
- Documentez clairement les attentes pour les classes qui héritent d'une classe parente ou qui implémentent une interface, surtout en ce qui concerne le comportement attendu des méthodes.

En suivant ce guide, vous pourrez utiliser le polymorphisme en PHP pour créer des applications plus modulaires, flexibles et maintenables.