

# Fiche de révision

## Les Classes Abstraites

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

Une classe abstraite en programmation orientée objet est une classe qui ne peut pas être instanciée directement et est destinée à servir de classe de base pour d'autres classes. Voici les caractéristiques principales des classes abstraites :

**Non instanciable** : Vous ne pouvez pas créer un objet directement à partir d'une classe abstraite. L'essayer provoquera une erreur.

**Héritage obligatoire** : Les classes abstraites sont conçues pour être étendues. Les classes enfants qui héritent d'une classe abstraite peuvent être instanciées, à condition qu'elles implémentent les méthodes abstraites définies dans la classe parente.

**Méthodes abstraites** : Une classe abstraite peut avoir des méthodes abstraites. Ce sont des méthodes déclarées sans implémentation et doivent être définies dans les classes enfants. Ces méthodes abstraites définissent un "contrat" que toutes les classes dérivées doivent suivre.

**Méthodes concrètes** : Une classe abstraite peut également contenir des méthodes concrètes avec une implémentation complète. Cela permet de partager un comportement commun entre toutes les classes enfants.

**Constructeurs** : Les classes abstraites peuvent avoir des constructeurs. Si une classe enfant est dérivée d'une classe abstraite avec un constructeur, elle doit appeler explicitement le constructeur de la classe parente, si nécessaire.

**Propriétés** : Tout comme les classes normales, les classes abstraites peuvent avoir des propriétés qui peuvent être héritées par des classes enfants.

En résumé, les classes abstraites en programmation orientée objet servent de squelette ou de modèle pour d'autres classes. Elles permettent de définir un ensemble de fonctionnalités et de contraintes que les classes dérivées doivent implémenter, favorisant ainsi une conception et une architecture logicielles cohérentes.

“Linux is only free if your time has no value”

## 2. Comment faire ?

Les classes abstraites en PHP sont un concept fondamental en programmation orientée objet (POO). Une classe abstraite est une classe qui ne peut pas être instanciée par elle-même et est destinée à être héritée par d'autres classes.

### 2.1. Déclarer une Classe Abstraite

Utilisez le mot-clé **abstract** pour déclarer une classe abstraite. Le mot-clé **protected** permet aux classes enfants d'accéder à l'attribut de la classe parent.

```
abstract class Vehicule {  
    protected $marque; // private empêcherait aux classes enfants d'accéder à l'attribut!  
  
    public function __construct($marque) {  
        $this->marque = $marque;  
    }  
}
```

### 2.2. Ajouter une méthode abstraite dans la classe parent

Les méthodes abstraites définissent la signature de la méthode sans implémentation. Les classes enfants sont responsables de fournir des implémentations pour ces méthodes.

```
abstract public function demarrer();
```

### 2.3. Hériter d'une Classe Abstraite

Une classe enfant hérite d'une classe abstraite en utilisant le mot-clé **extends**. La classe enfant doit fournir des implémentations pour toutes les méthodes abstraites de la classe parent.

```
class Voiture extends Vehicule {  
    public function demarrer() {  
        echo "La voiture " . $this->marque . " démarre.";  
    }  
}
```

### 2.4. Instancier la Classe Enfant

Vous ne pouvez pas instancier une classe abstraite directement. Vous devez instancier une classe enfant qui fournit des implémentations pour toutes les méthodes abstraites.

```
$maVoiture = new Voiture("Toyota");  
$maVoiture->demarrer(); // Affiche "La voiture Toyota démarre."
```

### 3. Bonnes Pratiques

Quelques bonnes idées à mettre en œuvre pour une bonne utilisation :

- **Utilisation Judicieuse** : Utilisez des classes abstraites lorsque vous avez un ensemble de classes qui partagent une structure ou un comportement commun, mais avec des implémentations différentes pour certaines méthodes.
- **Nommer Convenablement** : Nommez vos classes abstraites de manière à refléter leur usage ou le rôle dans la hiérarchie d'héritage.
- **Documentation** : Documentez clairement l'objectif de la classe abstraite et de chaque méthode abstraite pour guider les développeurs sur la manière de les étendre correctement.
- **Ne Pas Surcharger** : Évitez de mettre trop de méthodes dans une classe abstraite. Respectez le principe de responsabilité unique.