

I) Héritage

L'héritage fournit un couplage entre une classe parent et toute la classe fille. C'est une dépendance qui est codée en dur via le mot clé « extends ».

Avantages :

- Facile à créer des classes filles. La hiérarchie conçue est facile à étendre et de cette manière, l'ajout d'une nouvelle classe fille ne nécessite aucune modification du code existant
- Facile pour changer de comportement. De grands changements de comportement peuvent être obtenus via de petits changements dans le code existant.

Inconvénient :

- Souvent utiliser pour la duplication de code

Exemple :

```
<?php  
  
class car{}  
class Sportcar extends Car{}
```

II) Composition

Lorsque votre but principal est de réutiliser du code, la composition offre la possibilité d'utiliser des comportements commutables au moment de l'exécution.

Avantages :

- Changement d'interface de classe parent. Il est plus facile de modifier l'interface de classe parent qu'avec l'héritage.
- Changement d'interface de classe fille. Il est plus facile de modifier l'interface de classe front-end, ce qui n'est pas le cas avec l'héritage.
- Changer le comportement à un moment donné. Permet de retarder la création des objets jusqu'à ce qu'ils deviennent nécessaires.

Inconvénient :

- Méthodes fournies peuvent être mises dans le type dérivé.

Exemple :

```
<?php

interface Employee {}
class Person implements Employee{}
class Organisation implements Employee{}
```

III) Conclusion

Lorsque vous choisissez entre composition et héritage, vous devez déterminer la relation entre les classes. Si c'est une relation alors l'héritage convient puisqu'une classe enfant est une classe parent. Cependant, certaines situations nécessitent que différents objets jouent un rôle commun, c'est alors que la composition arrive.