

Programmation orientée objet

Cours 5 - Surcharge de méthodes
Polymorphisme et classes
abstraites

Surcharger des méthodes

- Lorsqu'on déclare des classes qui héritent d'une classe de base, il est souvent très utile de redéfinir des méthodes qui sont déjà présentes dans la classe de base mais qui devraient avoir une fonction plus spécifique dans la classe fille.
- Il suffit de redéfinir la même méthode, avec le même nom dans la classe fille que celui de la méthode présente dans la classe mère.
- Cette méthode sera dorénavant appelée à la place de celle de la classe mère.
- Il est aussi possible d'appeler une méthode de la classe mère à partir d'une classe fille même si cette classe fille possède une méthode possédant le même nom.
 - Pour ce faire, il faut utiliser le mot-clé parent, suivi de :: (*l'opérateur de résolution de portée*) et du nom de la méthode.
 - Ex : `parent::methode();`

Méthodes et classes finales

- Il est possible d'empêcher une méthode d'être surchargée en mettant le mot clé final devant.
- Il est aussi possible d'empêcher une classe d'être héritée en mettant le même mot clé devant le mot class lors de sa définition.
- Voir exemple.

Polymorphisme

- Dans les langages orientés-objet qui offrent un langage fortement typé, comme C++, le concept d'héritage amène avec lui le concept de polymorphisme.
- Le polymorphisme consiste à créer une collection d'objets, elle-même typée à une classe de base (qui pourrait être abstraite) et à remplir cette collection d'objets qui appartiennent à une ou plusieurs classes héritant de cette même classe de base.
- Ce concept est moins présent en PHP parce que les variables (et les collections et les tableaux) ne sont pas fortement typées.
- Toutefois, le concept peut tout de même être utilisé sans être aussi strict qu'en langage fortement typé.

Le polymorphisme en PHP

- On prendra avantage du polymorphisme en PHP lorsque, par exemple, on remplira un tableau d'éléments qui héritent tous de la même classe de base mais qui ne sont pas de la même classe.
- Il sera ensuite possible d'appeler la même méthode de tous ces objets s'ils héritent tous de la même classe, et qu'ils possèdent tous la même méthode.
- Voir exemple

Classes et méthodes abstraites

- Pour forcer tous les objets à redéfinir une méthode d'une classe, il est préférable d'utiliser le concept de classe abstraite.
- **IMPORTANT** : Une classe abstraite ne peut pas être instanciée.
 - Elle ne sert donc qu'à être un patron pour de futures classes qui hériteront de ses attributs et méthodes.
 - Elle est généralement accompagnée de méthodes abstraites.
- Une méthode abstraite n'est en fait qu'une signature de méthode, sans définition.
 - Lorsque cette méthode est définie au sein d'une classe abstraite, cela force les classes qui hériteront de celle-ci à la redéfinir (surcharger).
 - Voir exemple.
- N.B. Une classe qui a une méthode abstraite doit *absolument* être abstraite!