**Polymorphisme**

Le polymorphisme permet à une méthode d'adopter plusieurs "formes" dans des classes différentes, c'est-à-dire d'être redéfinie dans des classes filles : une méthode d'une classe fille ne contiendra pas le même code que la méthode portant le même nom dans la classe parente.

**Exemple**

Dans la classe parente vehicule, la fonction avancer() gèrera des kilomètres pour les classes filles voiture, bus ou camion, tandis qu'elle retournera des milles nautiques (*miles* en anglais) pour la classe fille bateau car elle y est codée de la sorte :

Classe fille voiture:

<?php

class voiture extends vehicule

{

public function avancer(int $nbKm2)

{

$this->$\_nbKm = $this->$\_nbKm + $nbKm2;

return $this->\_nbKm;

}

}

?>

Classe fille bateau :

<?php

class bateau extends vehicule

{

private $\_nbMiles = 0;

public function avancer(int $nbMiles)

{

$this->$\_nbMiles = $this->$\_nbMiles + $nbMiles2;

return $this->\_nbMiles;

}

}

Attention, cet exemple est volontairmeent très simpliste; dans la réalité c'est plus complexe : il existe en fait plusieurs types de polymorphisme qu'on rencontre selon le langage implémenté. En PHP, le polymorphisme se limite à la redéfinition (principe de surcharge) d'une méthode dans une classe fille.