

MISE EN PRATIQUE DE LA POO EN PHP 4/6

Auteur : Sébastien Inion

AU PROGRAMME DE CE COURS...

Dans cette quatrième partie, nous verrons :

- ✱Ce qu'est une **constante de classe**,

- ✱Ce qu'est une **méthode statique**

- ✱Ce qu'est un **attribut statique**.

Et nous mettrons en application sous la forme d'un exercice toutes les notions déjà vues jusqu'à maintenant.

Nous réaliserons : La classe **Voiture**.



Rappel : Abstraction



- On peut avoir **des classes abstraites**. Cela veut dire qu'elle ne peuvent servir à instancier un objet. Elles ont un rôle de modèle lors de l'héritage.
abstract class Personnage
- On peut avoir **des méthodes abstraites**. Dans la même idée une méthode abstraite à son corps vide. Elle oblige le développeur à écrire cette méthode lors de l'héritage.
abstract public function()

Les constantes de classe

- Comme le nom l'indique ces constantes appartiennent à la classe et non aux objets.
- On utilise l'opérateur de portée `::` pour y accéder
- Une constante de classe se déclare avec l'opérateur **const**
- Pour y accéder depuis l'intérieur de la classe on utilise **self::**
- Pour y accéder depuis l'extérieur de la classe on utilise **Classe::CONST**

<https://replit.com/@Sebastien11/Cours4POO#exempleConstanteDeClasse.php>

Les constantes de classe



```
<?php
class Personnage
{
    private $_force;
    private $_localisation;
    private $_experience;
    private $_degats;

    // Déclarations des constantes en rapport avec la force.

    const FORCE_PETITE = 20;
    const FORCE_MOYENNE = 50;
    const FORCE_GRADE = 80;

    public function __construct($forceInitiale)
    {
        // N'oubliez pas qu'il faut assigner la valeur d'un attribut uniquement depuis son
        // setter !
        $this->setForce($forceInitiale);
    }

    public function deplacer()
    {
    }

    public function frapper()
    {
    }

    public function gagnerExperience()
    {
    }

    public function setForce($force)
    {
        // On vérifie qu'on nous donne bien soit une « FORCE_PETITE », soit une « FORCE_MOYENNE
        // », soit une « FORCE_GRADE ».
        if (in_array($force, [self::FORCE_PETITE, self::FORCE_MOYENNE, self::FORCE_GRADE]))
        {
            $this->_force = $force;
        }
    }
}
```

Méthodes statiques

- Une méthode statique dépend de la classe et non du ou des objets.
- On peut l'utiliser sans créer d'objet.
Pour l'appeler de l'extérieur de l'objet : *Classe::methode()*
Exemple : **Personnage::parler()**
- Pour l'appeler de l'intérieur de l'objet :
self::parler()

Attributs statiques

- Un attribut statique dépend de la classe et non du ou des objets.
- Pour l'appeler de l'extérieur de l'objet :
*Exemple : **Personnage::\$monAttribut***
- Pour l'appeler de l'intérieur de l'objet :
self::\$monAttribut

Méthodes statiques

```
1  <?php
2  class Personnel {
3      protected static $NbPersonne ;
4
5      // -- Propriétés --
6      private $_nom;
7      private $_prenom;
8      private $_age;
9
10     public function __construct($n, $p, $a) {
11         $this->_nom = $n;
12         $this->_prenom = $p;
13         $this->_age = $a;
14         self::$NbPersonne++;
15     }
16
17     // Methode statique
18     static function NbrPersonne() {
19         return self::$NbPersonne;
20     }
21     // Methodes
22
23     public function AffichePersonne() {
24         echo self::$NbPersonne." Personne : ";
25         echo $this->_nom."\t".$this->_prenom."\t".$this->_age;
26         echo PHP_EOL;
27         echo "<br/>";
28     }
29 }
30
31 // Programme
32 $salarie = new Personnel("Combette","Roland",50);
33 $salarie1 = new Personnel("Ferreira","Antoine",32);
34 echo Personnel::NbrPersonne();
35 ?>
```


MISE EN APPLICATION...

Mise en application

Ecrire une classe **Vehicule** qui possède comme propriétés démarré (boolean), une vitesse actuelle et une vitesse maximum pour le véhicule. Cette classe à également 4 méthodes abstraites :

- demarrer()
 - eteindre()
 - ralentir(valeur)
 - accélérer(valeur)
-
- Construire une classe **Voiture** qui hérite de Véhicule.
Implémenter les méthodes en essayant d'être le plus près de la réalité.
Par exemple :
 - on ne peut accélérer si le moteur n'est pas allumé.
 - on ne peut ralentir de 30 km/h si on est à 20 km/h.
 - lorsqu'on accélère on ne peut dépasser la vitesse maximum.
 - implémenter la méthode de classe (méthode statique) qui donne le nombre de voiture instanciés. On suppose qu'on ne détruit pas de véhicule.
 - implémenter une constante de classe **VITESSE_MAX = 260** qui interdit d'instancier un véhicule avec une vitesse supérieur.
 - On peut interdire ou limiter l'accélération à 20 km/h par appel de fonction ou à 20 % de la vitesse courante.



**MERCI POUR
VOTRE ATTENTION**