

La Programmation Orientée Objet (POO) avec PHP

Objectifs

Deux objectifs sont définis pour cet atelier

1. Être en mesure d'identifier les principaux composants et mécanismes d'un objet
2. Comprendre et utiliser la programmation orienté objet et ses outils

Comment apprécier la POO

Avantages de la POO

1. Plus grande modularité car on isole dans un ensemble des fonctions (méthodes) et variables (propriétés) en fonction d'une fonction commune.
2. Plus aisé lorsque plusieurs programmeurs travaillent ensemble.
3. En comparaison au modèle procédurale (sur la base de fonctions), résout le problème de nomenclature des fonctions.

Inconvénients de la POO

1. Demande plus de ressources à l'exécution que le modèle procédural.

Références

Outils qui ont servi pour la construction de cet atelier

Le site Apprendre-php.com

<http://www.apprendre-php.com/tutoriels/tutoriel-29-les-classes-et-objets.html>

Cours de PHP 5 par Guillaume Rossolini sur Developpez.com

<http://g-rossolini.developpez.com/tutoriels/php/cours/?page=poo#LIV>

PHP 5 avancé (4^e édition) – Edition Eyrolles

Classe

Ensemble de propriétés communes défini par des propriétés qui représente l'état de l'objet et des méthodes qui représente sont comportement.

Propriétés

Sont des variables dans lesquels ont recense des informations qui pourront être utiles à l'exploitation (affichage, traitement, ...).

Méthode

Sont des fonctions qui traitent des opérations.

Accès des propriétés et méthodes

public : accessible depuis l'extérieur de la classe

private : accessible uniquement à l'intérieur de la classe

protected : accessible uniquement à la classe et aux classes parents

Déclaration de l'instance de la classe :

```
$instance = new MaClasse();
```

Accès à une propriété

```
$instance->vitesse ;
```

Accès à une propriété

```
$instance->get_distance() ;
```

`get_class` : Permet de visualiser la classe appelée.

`get_class_methods` : Permet de visualiser les methods publiques de la classe.

`get_class_vars` : Permet de visualiser les valeurs des propriétés publiques.

```
<?php
class MaClasse{

    public $vitesse = 50;
    private $distance = 200;
    public $tempsAlloue = 30;

    public function temps(){
        return round( $this->distance / $this->vitesse, 2 );
    }

    public function get_distance(){
        return $this->distance;
    }

    public function set_vitesse( $newVitesse ){
        $this->vitesse = $newVitesse;
    }

    public function heuresEnMinutes( $heures ){
        return $heures * 60;
    }

}

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
<head>
    <title>PHP -> POO</title>
</head>
<body>
<?php
$instance = new MaClasse();
echo $instance->vitesse.'km/h';
?>

<br />

<?php echo $instance->temps().' heures pour '.$instance->
get_distance().'km'; ?>

<br />

<?php echo $instance->heuresEnMinutes( $instance->temps() ).' minutes
pour '.$instance->get_distance().'km'; ?>

<br /><br />

<?php
$mvitesse = 200;
$instance->set_vitesse( $mvitesse );
echo $instance->heuresEnMinutes( $instance->temps() ).' minutes pour
'.$instance->get_distance().'km à '.$instance->vitesse.' km/h';
?>

<br /><br />

<?php
echo 'Classe : '.$get_class( $instance ).'<br />';

$methods = get_class_methods( $instance );
foreach( $methods as $method ){
    echo 'Méthode : '.$method.'<br />';
}

$vars = get_class_vars( 'MaClasse' );
foreach( $vars as $var ){
    echo 'Propriétés : '.$var.'<br />';
}
?>
</body>
```

__construct

Permet d'initialiser des valeurs qui se logeront dans les propriétés ciblées.

Elles se déclarent en même temps que lors de l'instanciation de l'objet.

```
<?php

class MaClasse{

    private $vitesse;
    public $distance;

    function __construct( $vitesse, $distance ){
        $this->distance = $distance;
        $this->vitesse = $vitesse;
    }

    function temps(){
        return round( $this->distance / $this->vitesse, 2 );
    }

}

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
<head>
    <title>PHP -> POO</title>
</head>
<body>
<?php
?>

<?php
$instance = new MaClasse( 25, 200 );
?>
<br />
<?php
echo $instance->temps(). ' heures pour ' . $instance->distance . 'km';
?>

</body>
```

```

CREATE TABLE IF NOT EXISTS
`membres` (
  `IdMembre` int(10) NOT NULL
  AUTO_INCREMENT,
  `NomMembre` varchar(50) NOT
  NULL,
  `PrenomMembre` varchar(50)
  NOT NULL,
  PRIMARY KEY (`IdMembre`)
) ENGINE=InnoDB DEFAULT
CHARSET=utf8
ROW_FORMAT=COMPACT
AUTO_INCREMENT=1 ;

```

```

class Membres{

    private $NomMembre = array();
    private $PrenomMembre = array();

    public function AllMembres(){

        $sql = 'SELECT * FROM membres';
        $result = mysql_query( $sql );
        while( $row = mysql_fetch_assoc( $result ) ){
            $this->NomMembre[] = $row['NomMembre'];
            $this->PrenomMembre[] = $row['PrenomMembre'];
        }

        $nbRows = mysql_num_rows( $result );

        return $this->MembresArray( $nbRows );
    }

    public function MembreFromId( $id ){

        $sql = 'SELECT * FROM membres WHERE IdMembre = \''.$id.'\'';
        $result = mysql_query( $sql );
        while( $row = mysql_fetch_assoc( $result ) ){
            $this->NomMembre[] = $row['NomMembre'];
            $this->PrenomMembre[] = $row['PrenomMembre'];
        }

        $nbRows = mysql_num_rows( $result );

        return $this->MembresArray( $nbRows );
    }

    public function MembresArray( $NbMembres ){

        $MembresArray = array();

        for( $i = 0; $i < $NbMembres; $i++ ){
            $array = array(
                'PrenomMembre' => $this->PrenomMembre[$i],
                'NomMembre' => $this->NomMembre[$i],
            );
            $MembresArray[] = $array;
        }
        return $MembresArray;
    }

    public function get_NomMembre(){
        return $this->NomMembre;
    }

    public function get_PrenomMembre(){
        return $this->PrenomMembre;
    }
}

?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
        <title>PHP->POO</title>
    </head>

    <body>
    <?php

        $AllMembres = new Membres();

        $membresArray = $AllMembres->AllMembres();
        foreach( $membresArray as $membreValue ){
            echo $membreValue['PrenomMembre'];
            echo ' ';
            echo $membreValue['NomMembre'];
            echo '<br />';
        }

        $membre = new Membres();

        $membreValue = $membre->MembreFromId( '2' );

        echo $membreValue[0]['PrenomMembre'];
        echo ' ';
        echo $membreValue[0]['NomMembre'];
        echo '<br />';

    ?>
    </body>

```