

Web (Persistence)



Andrea G. B. Tettamanzi

Université de Nice Sophia Antipolis

Département Informatique

andrea.tettamanzi@unice.fr

CM - Séance 7

Programmation orientée objet en PHP

Introduction

- Depuis PHP 5, le modèle objet a été réécrit : une des modifications majeures par rapport à PHP 4
- PHP 5 offre un modèle objet complet
- Parmi les fonctionnalités offertes par PHP 5, on peut citer
 - Visibilité dans les classes,
 - Classes « abstract » et « final »
 - Interfaces
 - Méthodes magiques
 - Clônage
 - typage (*typehinting*)
- Les variables sont des références vers des objets

Syntaxe de base

```
<?php
class SimpleClass
{
    // déclaration d'une propriété
    public $var = 'une valeur par défaut';

    // déclaration des méthodes
    public function displayVar() {
        echo $this->var;
    }
}
?>
```

Définition d'une classe ou interface

- Mot-clé « class » ou « interface »
- Héritage de réalisation : « extends »
 - Pas d'héritage multiple : une classe ne peut hériter que d'une seule classe parente (mais « traits » depuis PHP 5.4.0 !)
- Héritage d'interface : « implements » (interfaces : « extends »)
 - Héritage multiple permis : une classe peut réaliser plusieurs interfaces
- Constructeur : un seul (nombre de paramètres non discriminant)
 - Fonction magique « `__construct` »
 - Ou fonction au même nom de la classe (< PHP 5.3.3)
 - « `parent::__construct()` » pour appeler constructeur superclasse

Opérateur de résolution de portée

- Opérateur « :: », pa'anayim nekudotayim (פעמיים נקודתיים)
- Emprunté au C++
- Permet d'accéder aux membres « static » ou « constant », ainsi qu'aux propriétés ou méthodes surchargées d'une classe
- Utilisation :
 - `MaClasse::VALEUR_CONSTANTE`
 - `MaClasse::$variable_statique`
 - `MaClasse::méthode()`
 - `parent::__construct()`

Membres

- Une classe peut contenir ses propres constantes, variables (appelées « propriétés » ou « attributs »), et méthodes.
- Constantes : déclarées avec « const »
- Modificateur d'accès
 - private / protected / public
- Modificateur « static » pour déclarer attributs de classe
- Modificateur « final » pour empêcher l'écrasement
- « \$this » se réfère à l'objet courant
- « self » se réfère à la classe courante
- « parent » se réfère à la superclasse ou classe de base
- « __toString() » représentation en chaîne caractères de l'objet

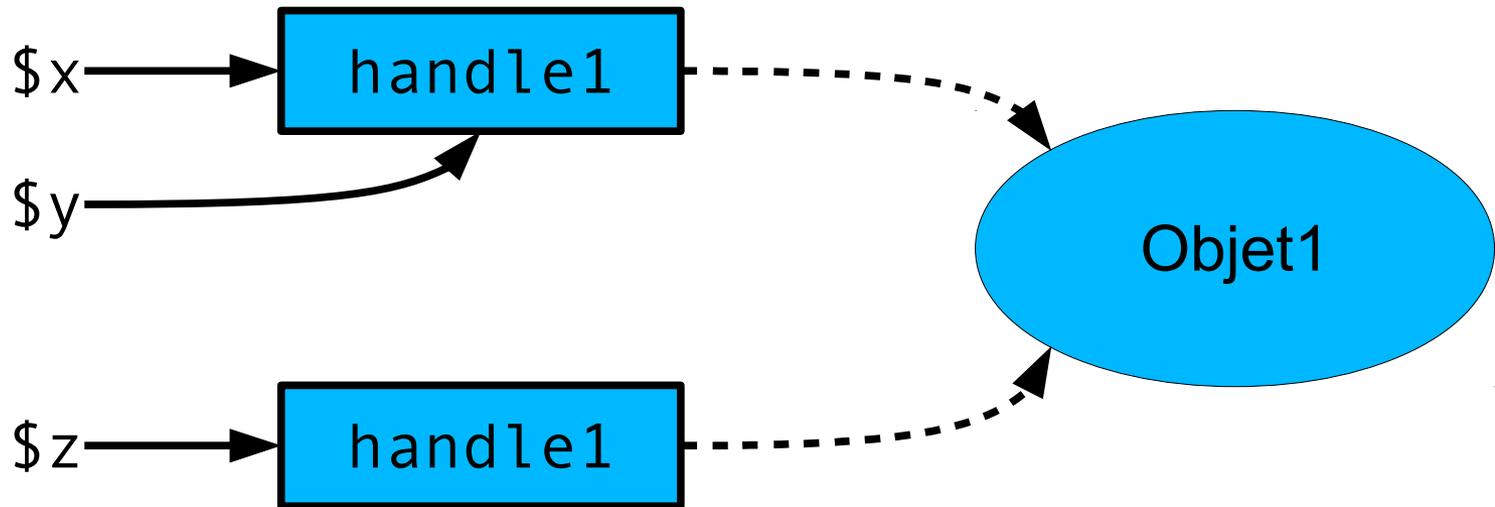
Création et utilisation

- Création
 - `$panier = new Panier();`
 - Le constructeur est la fonction qui est appelée automatiquement par la classe lorsque vous créez une nouvelle instance d'une classe à l'aide de l'opérateur `new`.
 - Le nom de la classe peut être dans une variable :
 - `$panier = new $nomClasse();`
- Accès aux membres d'une classe
 - `$panier->add_item("diplome", 1, 180);`
 - `$panier->add_item("semestre 5", 1, 30);`
 - `print_r($panier ->items);`
- Itération : « `foreach($objet as $key => $value) ... »`

Objets et références

- Un des piliers de la POO de PHP 5 est le fait que les "objets sont passés par référence par défaut". Ce n'est pas complètement vrai
- Une référence PHP est un alias, qui permet à deux variables différentes de représenter la même valeur.
- Depuis PHP 5, une variable objet ne contient plus l'objet en lui-même.
- Elle contient seulement un identifiant d'objet, qui permet aux accesseurs d'objets de trouver cet objet.
- Lorsque l'objet est utilisé comme argument, retourné par une fonction, ou assigné à une autre variable, les différentes variables ne sont pas des alias : elles contiennent des copies de l'identifiant, qui pointent sur le même objet.

Objets et références



```
$x = new Objet()
```

```
$y = &$x           # référence
```

```
$z = $x           # copie de la handle
```

Clonage

- Une copie d'objet est créée en utilisant le mot-clé clone (qui fait appel à la méthode « `__clone()` » de l'objet, si elle a été définie).
- La méthode « `__clone()` » ne peut être appelée directement.
- PHP effectue une copie superficielle des propriétés de l'objet.
- Toutes les propriétés qui sont des références à d'autres variables demeureront des références.
- Une fois le clonage effectué, si une méthode « `__clone()` » est définie, elle sera appelée, pour permettre à chaque propriété qui doit l'être d'être modifiée

Comparaison d'objets

- $\$a == \b
 - $\$a$ et $\$b$ ont les mêmes attributs et valeurs
 - $\$a$ et $\$b$ sont des instances de la même classe
- $\$a === \b
 - $\$a$ et $\$b$ font référence à la même instance de la même classe (c-à-d, leurs « handles » sont identiques)

Type Hinting

- = « allusion à un type »
- PHP 5 introduit le typage explicite
- Les fonctions peuvent maintenant imposer aux paramètres un typage (en spécifiant le nom de la classe dans le prototype de la fonction), et ainsi d'être des Objets, des interfaces, des tableaux (depuis PHP 5.1), ou de type callable (depuis PHP 5.4).
- Si une classe ou une interface est spécifiée comme type, alors toutes les classes en héritant ou ses implémentations seront aussi autorisées.
- La spécification d'un type ne peut être utilisée avec les types scalaires comme int ou string. Les ressources et les traits ne sont pas non plus autorisés.

Sérialisation d'objets

- Sérialisation : `serialize()`
- Désérialisation : `unserialize()`
- Noter que les membre statiques ne sont pas sérialisés

Merci de votre attention

